

**MACMHA - Marco Conceptual De Atributos, Métricas Y Heurísticas De
Calidad De Software Para La Valoración
Del Producto Software Orientado A Objetos**

ANEXOS



**Liliam Paola Bolaños Rengifo
Manuel Alejandro Navia Porras**

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
PROGRAMA INGENIERÍA DE SISTEMAS
Grupo de I+D en Tecnologías de la Información
POPAYÁN
2010**

**MACMHA - Marco Conceptual De Atributos, Métricas Y Heurísticas De
Calidad De Software Para La Valoración
Del Producto Software Orientado A Objetos**

ANEXOS



**Liliam Paola Bolaños Rengifo
Manuel Alejandro Navia Porras**

Trabajo de investigación para optar al título de Ingenieros de Sistemas

Director:

Mag. Jorge Jair Moreno Chaustre

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
PROGRAMA INGENIERÍA DE SISTEMAS
Grupo de I+D en Tecnologías de la Información
POPAYÁN
2010**

TABLA DE CONTENIDO

ANEXO A: APLICACIÓN DE LOS CRITERIOS A LOS MODELOS DE CALIDAD DE SOFTWARE	5
ANEXO B: ATRIBUTOS VS MODELOS DE CALIDAD PARA EL PRODUCTO SOFTWARE	16
ANEXO C: APLICACIÓN DE LOS CRITERIOS A LOS ATRIBUTOS DE CALIDAD ENCONTRADOS.....	20
ANEXO D: APLICACIÓN DE LOS CRITERIOS A LAS MÉTRICAS DE CALIDAD ENCONTRADAS.....	24
ANEXO E: APLICACIÓN DE LOS CRITERIOS A LAS HEURÍSTICAS DE CALIDAD ENCONTRADAS.....	41
ANEXO F: RELACIÓN ATRIBUTOS-HEURÍSTICAS DE CALIDAD SOFTWARE.....	47
ANEXO G: ATRIBUTOS DE CALIDAD ENCONTRADOS.....	54
ANEXO H: LAS MÉTRICAS DE CALIDAD ENCONTRADAS	97
ANEXO I: LAS HEURÍSTICAS DE CALIDAD ENCONTRADAS	334
ANEXO J: INFLUENCIAS.....	376
ANEXO K: RECOMENDACIONES POR DISCIPLINA.....	380
ANEXO L: SONDEO DE EMPRESAS DE DESARROLLO DE SOFTWARE EN EL SUR OCCIDENTE COLOMBIANO.....	388
<i>L1. Formato de la encuesta.....</i>	388
<i>L2. Resultados de la empresa QHATU.....</i>	390
<i>L3. Resultados empresa Latin Business</i>	391
<i>L4. Resultados empresa SITI</i>	392
<i>L5. Resultados empresa SERATIC.....</i>	393
<i>L6. Resultados empresa INPUT.....</i>	395
<i>L7. Resultados generales.....</i>	396
ANEXO M: ENCUESTA PARA EL DIAGNOSTICO RESPECTO A PRACTICAS DEL PRODUCTO SOFTWARE.....	397
<i>M1. Formato de la encuesta.....</i>	397
<i>M2. Resultados de la empresa QHATU.....</i>	399
<i>M3. Resultados empresa Latin Business</i>	400

M4.	Resultados empresa SITI	402
M5.	Resultados empresa SERATIC	403
M6.	Resultados empresa INPUT	404
M7.	Resultados generales	406
ANEXO N: ENCUESTA: INSPECCIÓN DE LAS PRÁCTICAS APLICADAS AL PRODUCTO SOFTWARE.....		407
N1.	Formato de la encuesta.....	407
N2.	Resultados de la empresa QHATU.....	409
N3.	Resultados empresa Latin Business	410
N4.	Resultados empresa SITI	412
N5.	Resultados empresa SERATIC	413
N6.	Resultados empresa INPUT	414
N7.	Resultados generales.....	416
ANEXO O: ENCUESTA: INSPECCIÓN DE FUENTES USADAS POR LAS EMPRESAS DE LA REGIÓN PARA LA CALIDAD DEL PRODUCTO SOFTWARE		417
O1.	Formato de la encuesta.....	417
O2.	Resultados de la empresa QHATU.....	419
O3.	Resultados empresa Latin Business	420
O4.	Resultados empresa SITI	421
O5.	Resultados empresa SERATIC	422
O6.	Resultados empresa INPUT	423
O7.	Resultados generales.....	424
ANEXO P: ENCUESTA DE CONVENIENCIA EN EL USO DE ATRIBUTOS DE CALIDAD PARA VALORAR EL PRODUCTO SOFTWARE		426
P1.	Diseño de la Encuesta.....	426
P2.	Resultados de la Empresa SERATIC.....	429
P3.	Resultados de la Empresa INPUT TECHNOLOGIES.....	431
P4.	Resultados de la Empresa LATIN BUSINESS	434
P5.	Resultados de la Empresa SITI.....	436
P6.	Resultados generales de las empresas.....	439
ANEXO Q: APLICACIÓN DEL MARCO EN LAS EMPRESAS		444

Q1.	<i>Selección de atributos y métricas.....</i>	444
Q2.	<i>Resultados de la Empresa SERATIC.....</i>	444
Q3.	<i>Resultados de la Empresa INPUT TECHNOLOGIES.....</i>	445
Q4.	<i>Resultados de la Empresa LATIN BUSINESS</i>	445
Q5.	<i>Resultados de la Empresa SITI.....</i>	446
Q6.	<i>Resultados generales de las empresas.....</i>	447
ANEXO R:	ENCUESTA DE SATISFACCIÓN	452
R1.	<i>Diseño de la Encuesta.....</i>	452
R2.	<i>Resultados de la Empresa SERATIC.....</i>	453
R3.	<i>Resultados de la Empresa INPUT TECHNOLOGIES.....</i>	453
R4.	<i>Resultados de la Empresa LATIN BUSINESS</i>	454
R5.	<i>Resultados de la Empresa SITI.....</i>	455
ANEXO S:	INFORME.....	457
ANEXO T:	PLANTILLAS PARA LA EVALUACIÓN DEL PRODUCTO SOFTWARE ...	466
ANEXO U:	REFERENCIAS	468
ANEXO V:	SEGUIMIENTO DE ACTIVIDADES.....	483
ANEXO W:	ARTÍCULOS PUBLICADOS.....	485
W1.	<i>Artículo: Un acercamiento a las prácticas de calidad de software en las MiPyMESPS^s del Sur-occidente colombiano*</i>	485
W2.	<i>Artículo: Breve Exploración de Modelos y Estándares de Calidad para el Producto Software</i>	498
ANEXO X:	CARTAS DE CERTIFICACION DE ASISTENCIA A LAS EMRESAS	511
ANEXO Y:	CARTAS DE RECEPCION DE ARTICULOS	515

ANEXO A: APLICACIÓN DE LOS CRITERIOS A LOS MODELOS DE CALIDAD DE SOFTWARE

El propósito de este Anexo es presentar los resultados de la valoración realizada a cada uno de los modelos de calidad software presentados en el capítulo 1, además de incluir su respectiva justificación y Calificación.

Criterios	Modelo de McCall [18]	Calificación
	Justificación	
C1: Disponibilidad	La información sobre el modelo de McCall [18], esta en distintos textos académicos, como libros, artículos, trabajos de tesis de grado, etc., sin embargo el texto en donde el modelo esta es difícil de encontrar además de tener un costo comercial elevado. Los documentos de que refieren al documento principal no siempre son exactos y en algunas ocasiones son limitados, ya sea por la información que presentan o el costo.	2
C2: Claridad	La documentación del modelo es presentada en tres reportes técnicos, en los cuales describe una serie de aspectos a seguir para evaluar la calidad de software, pero no es claro en cómo debe ser usado el modelo a pesar de que en su tercer volumen dice ser una guía de uso, el modelo esta escrito en inglés.	2
C3: Adaptabilidad	Pensado como un modelo jerárquico que sea capaz de adaptarse a las necesidades del usuario del modelo. Requiere de seguir ciertos pasos para poder ser aplicado.	2
C4: Completitud	A pesar de que no es del todo claro describe la mayoría de aspectos importantes, como atributos y métricas.	2
C5: Área de aplicación	Es un modelo presentado para evaluar a calidad de productos terminados, o hacer evaluación en hitos de el proyecto	3

Tabla 1. Calificación del Modelo de McCall [18]

Criterios	Modelo de Boehm [17]	Calificación
	Justificación	
C1: Disponibilidad	Al igual que el modelo de McCall [18], la información de el modelo de Boehm [17] tiene limitaciones de costo y acceso, la información puede encontrarse en diversos textos académicos en forma limitada	2
C2: Claridad	El modelo es documentado claramente en sus	2

	atributos, pero cuando llega al método de medición es confuso por la presentación de sus métricas, no es claro como debe hacerse la evaluación del producto. El modelo esta presentado en inglés. No posee una guía de uso.	
C3: Adaptabilidad	Requiere uso de unas ciertas características para poder ser aplicado correctamente, sin embargo permite escoger entre sus opciones presentadas.	2
C4: Completitud	No es un modelo que sea presentado de forma completa, si bien ha registrado una serie de atributos y métricas no posee material de apoyo.	2
C5: Área de aplicación	Muestra ser un modelo hecho para evaluar la calidad del producto software.	3

Tabla 2. Calificación del Modelo de Boehm [17]

Criterios	<i>Modelo de FURPS [20]</i>	Calificación
	Justificación	
C1: Disponibilidad	FURPS [20], por ser un modelo inicialmente diseñado para la compañía HP, tiende a no presentar información suficiente al público en general, los costos de adquisición del material son elevados, y no es fácil de encontrar información relevante.	1
C2: Claridad	Para este ítem el modelo es claro pero como no hay suficiente documentación es difícil saber si presentan el cómo debe ser usado.	2
C3: Adaptabilidad	Es bastante flexible a las necesidades de uso del modelo, desde un comienzo recomienda usar lo que realmente es deseado y necesitado a aplicar.	3
C4: Completitud	Es un modelo incompleto pues no presenta la totalidad de información necesaria para poder usarlo.	1
C5: Área de aplicación	Es un modelo para evaluar la calidad del producto software, principalmente pensado para la producción de HP.	3

Tabla 3. Calificación del Modelo FURPS [20]

Criterios	<i>Modelo de GILB [4]</i>	Calificación
	Justificación	
C1: Disponibilidad	El modelo de GILB [4], presenta información en la página de su creador, sin embargo es limitada la información encontrada sobre estudios realizados por otros autores acerca de este modelo.	2
C2: Claridad	El acceso a la información fue limitado, y no fue posible determinar este criterio.	NA

C3: Adaptabilidad	No fue posible conocer el modelo a fondo.	NA
C4: Completitud	No fue posible conocer el modelo a fondo.	NA
C5: Área de aplicación	Es un modelo pensado para la especificación de requisitos de software	3

Tabla 4. Calificación del Modelo de Gilb [4]

Criterios	<i>Estándar IEEE 610.12 [23]</i>	Calificación
	Justificación	
C1: Disponibilidad	Estadisponible en la página de IEEE en forma digital, sin embargo posee un costo comercial.	2
C2: Claridad	Debido a que no es un modelo este criterio no Aplica.	NA
C3: Adaptabilidad	Debido a que no es un modelo este criterio no Aplica.	NA
C4: Completitud	Debido a que no es un modelo este criterio no Aplica.	NA
C5: Área de aplicación	Es un estándar dedicado a ser glosario de términos para la calidad de software.	1

Tabla 5. Calificación del Estandar IEEE 610.12 [23]

Criterios	<i>Modelo de SATC [83]</i>	Calificación
	Justificación	
C1: Disponibilidad	Existen varios artículos que hacen mención de este modelo, además la información de este puede ser descargada de la página de la NASA (www.nasa.gov). Limitado a estudios realizados por los creadores	2
C2: Claridad	El modelo estadocumentado en forma entendible, y aunque presenta algunos ejemplos de cómo debe ser usado, no es del todo claro en qué aspectos puede hacerse variaciones del mismo o su modo de uso.	2
C3: Adaptabilidad	Visto como un modelo flexible para ser adaptado a los requerimientos de aplicación, sin embargo exige seguir ciertas normas.	2
C4: Completitud	Estadocumentado	3
C5: Área de aplicación	Es una metodología planteada para la calidad de los proyectos, incluye aspectos de calidad de requerimientos y producto, Efectividad de implementación y prueba.	2

Tabla 6. Calificación del Modelo SATC [83]

Criterios	<i>Modelo de Dromey [26]</i>	Calificación
	Justificación	
C1: Disponibilidad	Por ser un modelo relativamente nuevo los estudios sobre el mismo son limitados, aun así el	1

	autor ha realizado diversos artículos sobre el desarrollo del modelo. El problema es que el documento definitivo no ha sido publicado aun.	
C2: Claridad	En los documentos encontrados el modelo explica detalladamente su estructura, sin embargo no es claro el cómo debe ser usado, o su forma de medición.	2
C3: Adaptabilidad	No es del todo claro como debe ser usado el modelo, sin embargo sugiere que pueden tomarse solo las partes necesarias a ser aplicadas	2
C4: Completitud	No tiene la documentación completa de métricas y guía de uso. No está disponible	2
C5: Área de aplicación	Es un modelo para producto que busca incorporar aspectos adicionales a modelos ya planteados en el pasado como McCall [18], Boehm [17] e ISO 9126 [15].	3

Tabla 7. Calificación del Modelo de Dromey [26]

Criterios	Estándar ISO 14598 [27]	Calificación
	Justificación	
C1: Disponibilidad	Siendo una norma que acompaña el juego de normas de ISO 9126 [15], no es fácil de acceder debido a su costo.	2
C2: Claridad	Su estructura es clara en sí, sin embargo el uso de este en combinación con otros estándares es confuso.	1
C3: Adaptabilidad	Debido a que no es claro como debe ser usado el estándar este criterio Aplica.	NA
C4: Completitud	Debido a que no es un modelo para producto precisamente, este criterio no Aplica.	NA
C5: Área de aplicación	Ayuda a la evaluación del producto software, debe trabajarse en conjunto con ISO 9126 [15]:1 y solo su sección 1 está dedicada al producto como tal, desde su sección 2 hasta 6 esta designada al proceso y documentación del software.	1

Tabla 8. Calificación del Estándar ISO 14598 [27]

Criterios	Modelo de SQAE [30]	Calificación
	Justificación	
C1: Disponibilidad	Es un modelo desarrollado por MITRE, aunque son encontrados documentos donde describe el modelo y la información encontrada esta limitada a las experiencias de sus creadores.	2
C2: Claridad	Presenta documentación limitada, pero aquella encontrada es clara. No presenta una guía de	2

	cómo debe aplicarse el modelo	
C3: Adaptabilidad	Es modelo medianamente adaptable, pero exige la secuencia de algunos pasos.	2
C4: Completitud	Esta incompleta pues no presenta las métrica explícitamente, ni tampoco una guía de uso	2
C5: Área de aplicación	Es más considerada una metodología, sin embargo ayuda con la medición de la calidad para los productos software, en busca de calcular riesgos.	3

Tabla 9. Calificación del Modelo SQA [30]

Criterios	<i>Modelo de Bansiya [31]</i>	Calificación
	Justificación	
C1: Disponibilidad	Es un modelo relativamente nuevo, y existen algunos estudios relacionados, sin embargo el acceso al trabajo es limitado debido a costos.	2
C2: Claridad	El modelo presenta su estructura de forma comprensible para personas expertas, pero no es claro como es llevado el proceso de medición. Además que el uso de niveles agrega Complejidad a la hora de entenderlo	2
C3: Adaptabilidad	No es del todo claro cómo usar este modelo, sin embargo parece ser medianamente flexible a la hora de aplicarlo.	2
C4: Completitud	No tiene una guía de uso, y documenta su proceso incluyendo los atributos y métricas.	2
C5: Área de aplicación	Es un modelo de calidad para producto software orientado a objetos para la evaluar la calidad de atributos de alto nivel.	3

Tabla 10. Calificación del Modelo de Bansiya [31]

Criterios	<i>Modelo de GQM [8][32]</i>	Calificación
	Justificación	
C1: Disponibilidad	La información sobre este modelo puede encontrarse relativamente fácil, sin embargo hay discrepancias entre algunos autores lo que hace difícil recolectar información precisa.	2
C2: Claridad	Sigue una serie de pasos pero no es lo suficientemente claro para el usuario inexperto.	1
C3: Adaptabilidad	Es un modelo bastante adaptable, desde un comienzo el usuario define que es lo que va a medir y como lo hará.	3
C4: Completitud	A pesar de documenta su proceso no hay claridad en definición de atributos, métricas y guías de uso.	1
C5: Área de aplicación	Es un modelo de calidad para producto software, basado en los objetivos a lograr por el producto y	2

	proceso.	
--	----------	--

Tabla 11. Calificación del Modelo GQM [8] [32]

Criterios	<i>Estándar IEEE 1061 [16]</i>	Calificación
	Justificación	
C1: Disponibilidad	El estándar puede conseguirse en la página de IEEE, tiene costo comercial	2
C2: Claridad	Debido a que no es un modelo este criterio no Aplica.	NA
C3: Adaptabilidad	Debido a que no es un modelo este criterio no Aplica.	NA
C4: Completitud	Debido a que no es un modelo este criterio no Aplica.	NA
C5: Área de aplicación	Es una metodología para el establecimiento de los requerimientos de calidad y su identificación, implementación, análisis y validación del proceso y el producto de software. Pero no es presentado como un modelo de producto.	1

Tabla 12. Calificación del Estándar IEEE 1061 [16]

Criterios	<i>Modelo de ISO 9126 [15]</i>	Calificación
	Justificación	
C1: Disponibilidad	El modelo puede adquirirse por un costo comercial realmente elevado, en la página de ISO, y en Colombia lo comercializa ICONTEC. Varios autores lo han tomado como base para desarrollar sus trabajos, sin embargo no son conocidas publicaciones sobre resultados de experiencias de uso.	2
C2: Claridad	No es del todo claro como debe ser usado en modelo completo, en la parte 4 muestra una guía de uso, que un usuario inexperto podría no entender rápidamente.	2
C3: Adaptabilidad	Es un modelo medianamente adaptable. Presenta una serie de aspectos a seguir para alcanzar la calidad del producto, y permite usar aquellas partes que el usuario necesite aplicar	2
C4: Completitud	Es uno de los modelos más completos que hay, define atributos y métricas, además de presentar una guía de uso en su parte 4. Sin embargo, no posee documentos de ayuda para los datos de medición.	2
C5: Área de aplicación	Es un modelo para evaluar la calidad del producto software, presenta una jerarquía de atributos de calidad, dividiéndolos en internos y externos.	3

Tabla 13. Calificación del Modelo ISO 9126 [15]

Criterios	<i>Modelo de QUINT2 [58]</i>	Calificación
	Justificación	
C1: Disponibilidad	El acceso al modelo de Quint 2 esta limitado a unos pocos trabajos y la página de internet del mismo, sin embargo el inconveniente más grave de este modelo es el idioma en el que fue desarrollado.	1
C2: Claridad	Igual a ISO 9126 [15], no es del todo claro en cómo debe ser usado, además esta agregado el problema del idioma del documento, pues este modelo ha sido escrito en Alemán	2
C3: Adaptabilidad	Usado igual que ISO 9126 [15]	2
C4: Completitud	Estaincompleto pues esta limitado a su base conceptual en ISO 9126 [15]	2
C5: Área de aplicación	Es un modelo para producto software, planteado como un plus de ISO 9126 [15], para productos web.	3

Tabla 14. Calificación del Modelo QUINT2 [58]

Criterios	<i>Modelo PQM</i>	Calificación
	Justificación	
C1: Disponibilidad	Acera de PQM muchos textos, mencionan su estructura y forma de uso, es un modelo bien documentado, sin embargo en algunos casos la información relevante tiene costo comercial.	2
C2: Claridad	Debido a su forma de estructurarse puede ser complicado para un usuario inexperto, sin embargo cuando es utilizado es más fácil entender su proceso.	1
C3: Adaptabilidad	Al igual que GQM [8][32] es adaptable desde el principio	3
C4: Completitud	A pesar de documenta su proceso no hay claridad en definición de atributos, métricas y guías de uso	1
C5: Área de aplicación	Es un modelo dedicado a calcular la calidad de software para portales, basado en el modelo GQM [8][32]	2

Tabla 15. Calificación del Modelo PQM [4][35]

Criterios	<i>Modelo de Six Sigma</i>	Calificación
	Justificación	
C1: Disponibilidad	La información sobre Six Sigma, generalmente tiene costo comercial, aun así muchos autores han utilizado esta metodología.	2
C2: Claridad	Debido a que no es un modelo este criterio no Aplica.	NA

C3: Adaptabilidad	Debido a que no es un modelo este criterio no Aplica.	NA
C4: Completitud	Debido a que no es un modelo este criterio no Aplica.	NA
C5: Área de aplicación	Modelo para evaluación del proceso software	1

Tabla 16. Calificación de Six Sigma [4][21]

Criterios	<i>Estándar ISO 9000-3 (TickIT)</i>	Calificación
	Justificación	
C1: Disponibilidad	Es un modelo para calidad muy bien documentado, sin embargo tiene costo comercial, estadisponible en la página de ISO.	2
C2: Claridad	No pudo encontrarse información suficiente para llenar el criterio.	NA
C3: Adaptabilidad	Es un modelo medianamente rígido en su aplicación	2
C4: Completitud	No pudo encontrarse información suficiente para llenar este criterio	NA
C5: Área de aplicación	Modelo para evaluación del proceso software	1

Tabla 17. Calificación del Estándar ISO 9000-3 (TickIT) [21][37]

Criterios	<i>Modelo de ISO 15504 (SPICE)</i>	Calificación
	Justificación	
C1: Disponibilidad	Es un modelo para calidad muy bien documentado, muchos autores han hecho referencia y uso de él, sin embargo tiene costo comercial, y los resultados de uso sido poco documentados, estadisponible en la página de ISO.	2
C2: Claridad	Es suficientemente claro, sin embargo posee una Complejidad mayor que otros modelos	2
C3: Adaptabilidad	Es rígido en su aplicación.	1
C4: Completitud	No pudo encontrarse información suficiente para llenar este criterio	NA
C5: Área de aplicación	Modelo para evaluación del proceso software	1

Tabla 18. Calificación del Modelo SPICE [38]

Criterios	<i>Modelo de ISO 12207 o IEEE / EIA 12207</i>	Calificación
	Justificación	
C1: Disponibilidad	La norma peruana de iso 12207, estaen la página de ISO y en la de IEEE, con un costo comercial	2
C2: Claridad	Estabien documentado, pero no posee guías de	2

	ayuda para saber cómo debe aplicarse.	
C3: Adaptabilidad	Debido a la información que comparte con otros estándares es medianamente rígido en su aplicación.	2
C4: Completitud	Debido a que no es un modelo pensado para producto este criterio no Aplica.	NA
C5: Área de aplicación	Modelo para evaluación y mejora de los procesos software	1

Tabla 19. Calificación del Modelo ISO 10207 [4]

Criterios	<i>Modelo de PSP</i>	Calificación
	Justificación	
C1: Disponibilidad	Es una metodología poco utilizada por los autores. No hay referencias suficientes.	1
C2: Claridad	Debido a que no es un modelo este criterio no Aplica.	NA
C3: Adaptabilidad	Debido a que no es un modelo este criterio no Aplica.	NA
C4: Completitud	Debido a que no es un modelo este criterio no Aplica.	NA
C5: Área de aplicación	Mas que un modelo es un proceso a seguir por los ingenieros de software en forma individual.	1

Tabla 20. Calificación del Modelo PSP [4]

Criterios	<i>Modelo de PSM</i>	Calificación
	Justificación	
C1: Disponibilidad	Es una metodología poco utilizada por los autores. No hay referencias suficientes.	1
C2: Claridad	Debido a que no es un modelo este criterio no Aplica.	NA
C3: Adaptabilidad	Debido a que no es un modelo este criterio no Aplica.	NA
C4: Completitud	Debido a que no es un modelo este criterio no Aplica.	NA
C5: Área de aplicación	Proceso de medición orientado a la información.	1

Tabla 21. Calificación del Modelo PSM [38]

Criterios	<i>Modelo de Bootstrap</i>	Calificación
	Justificación	
C1: Disponibilidad	Es un modelo con referencias limitadas.	1
C2: Claridad	No pudo encontrarse información suficiente para llenar este criterio.	NA
C3: Adaptabilidad	No pudo encontrarse información suficiente para llenar este criterio.	NA
C4: Completitud	No pudo encontrarse información suficiente para llenar este criterio.	NA

C5: Área de aplicación	Modelo para evaluación del proceso software	1
------------------------	---	---

Tabla 22. Calificación del Modelo Bootstrap [37]

Criterios	<i>Modelo de TSP</i>	Calificación
	Justificación	
C1: Disponibilidad	Es una metodología poco utilizada por los autores. No hay referencias suficientes.	1
C2: Claridad	Debido a que no es un modelo este criterio no Aplica.	NA
C3: Adaptabilidad	Debido a que no es un modelo este criterio no Aplica.	NA
C4: Completitud	Debido a que no es un modelo este criterio no Aplica.	NA
C5: Área de aplicación	Es una especialización de PSP, pero para equipos de trabajo.	1

Tabla 23. Calificación del Modelo TSP [4]

Criterios	<i>Modelo de CMMI</i>	Calificación
	Justificación	
C1: Disponibilidad	Es un modelo muy bien documentado y con muchos autores que lo referencian, el modelo como tal tiene un costo comercial, sin embargo debido a la gran cantidad información encontrada puede lograrse una idea de su funcionamiento.	2
C2: Claridad	Es un modelo muy bien documentado, y posee una madurez suficiente	3
C3: Adaptabilidad	Es un modelo es caracterizado por qué, debe ser seguido estrictamente para alcanzar los niveles que plantea.	1
C4: Completitud	Es un modelo bastante completo en sus características internas, sin embargo como no es pensado precisamente para producto.	1
C5: Área de aplicación	Modelo para evaluación y mejora en el Mantenimiento de productos y servicios software	1

Tabla 24. Calificación del Modelo CMMI [21]

Criterios	<i>Estándar ISO 90003</i>	Calificación
	Justificación	
C1: Disponibilidad	El estándar puede encontrarse en la página de ISO, con un costo comercial.	2
C2: Claridad	Como ha sido plasmado presenta una claridad mayor a sus predecesores de ISO 9000, sin embargo no es exactamente un modelo.	NA
C3: Adaptabilidad	No pudo encontrarse información, en donde justifique su adaptabilidad.	NA
C4: Completitud	No pudo encontrarse información, en donde	NA

	justifique su adaptabilidad.	
C5: Área de aplicación	Provee una guía en adquisición, suministro, desarrollo, operación y Mantenimiento de software y servicios de soporte	1

Tabla 25. Calificación del Modelo ISO 90003 [4]

Criterios	<i>Modelo de ISO 25000 (SQUARE)</i>	Calificación
	Justificación	
C1: Disponibilidad	Un modelo muy referenciado, sin embargo tiene un costo comercial, esta en la página de ISO.	2
C2: Claridad	Está bien documentado en relación a sus prácticas de mejora de la calidad, sin embargo el estar relacionado con otros modelos lo hace confuso	2
C3: Adaptabilidad	Es un modelo medianamente rígido, lo que lo hace más adaptable es su conexión con otros estándares	2
C4: Completitud	No pudo encontrarse información, en donde justifique su adaptabilidad.	NA
C5: Área de aplicación	Trata de un conjunto de modelos de referencia para soportar los requerimientos de ISO 9126 [15] e ISO 14598 [27].	1

Tabla 26. Calificación del Modelo ISO 25000 [42]

Criterios	<i>Modelo de ISO 20000</i>	Calificación
	Justificación	
C1: Disponibilidad	Un modelo que tiene un costo comercial, esta en la página de ISO. Varios autores lo han mencionado, sin tener información completa.	2
C2: Claridad	Está bien documentado, sin embargo, no es claro cómo aplicarlo al producto software directamente.	NA
C3: Adaptabilidad	No pudo encontrarse información suficiente para llenar este criterio.	NA
C4: Completitud	No pudo encontrarse información suficiente para llenar este criterio.	NA
C5: Área de aplicación	Modelo para la especificación de administración de servicios para mejorar sus entregas a los usuarios.	1

Tabla 27. Calificación del Modelo ISO 20000 [4]

ANEXO B: ATRIBUTOS VS MODELOS DE CALIDAD PARA EL PRODUCTO SOFTWARE

Éste anexo muestra la frecuencia de aparición de los atributos de calidad, presentes en cada modelo o estándar para producto software.

Para mayor Facilidad es usada la siguiente notación; (ver Tabla 28)

MODELO/ESTÁNDAR vs ATRIBUTOS	McCall	Boehm	FURPS	Dromey	SQAE	Bansiya	ISO 9126	Quint 2	Frecuencia atributo
Abstracción						x			1
Abstracto				x					1
Accesibilidad		x							1
Acoplamiento						x			1
Adaptabilidad			x				x	x	3
Ajustable				x					1
Amigabilidad								x	1
Analizabilidad							x	x	2
Asignado				x					1
Atractivo							x	x	2
Auto descripción	x	x		x	x				4
Ayuda								x	1
Capacidad de crecimiento		x							1
Capacidades			x						1
Características			x						1
Claridad								x	1
Co existencia							x		1
Cohesión				x		x			2
Compatibilidad			x						1
Compatibilidad			x						1
Complejidad						x			1
Completitud	x	x		x					3
Comportamiento en el tiempo							x	x	2
Composición						x			1
Computable				x					1
Comunicatividad	x	x							2
Concisión	x	x							2
Confiabilidad	x	x	x	x			x	x	6
Configurabilidad			x						1
Conformidad								x	1
Conformidad con la Confiabilidad							x		1
Conformidad con la Eficiencia							x		1
Conformidad con la Funcionalidad							x		1
Conformidad con la Mantenibilidad							x		1

MODELO/ESTÁNDAR vs ATRIBUTOS	McCall	Boehm	FURPS	Dromey	SQAE	Bansiya	ISO 9126	Quint 2	Frecuencia atributo
Conformidad con la Usabilidad							x		1
Conformidad con Portabilidad							x		1
Consistencia	x	x	x		x				4
Consistente				X					1
Consumo de recursos			x						1
Contextual									0
Control de acceso	x								1
Control de anomalías					x				1
Corrección	x								1
Degradabilidad								x	1
Descripción					x				1
Descriptiva				X					1
Desempeño			x						1
Directo				x					1
Diseño simple				x	x				2
Disponibilidad								x	1
Documentación			x		x				2
Documentado				X					1
Efectividad						x			1
Efectivo				X					1
Eficiencia	x	x	x	X			x	x	6
Eficiencia de almacenamiento	x								1
Eficiencia de dispositivo		x							1
Eficiencia de ejecución	x								1
Encapsulamiento				x		x			2
Comprensibilidad		x				x	x	x	4
Entrenamiento	x								1
Especificado				x					1
Estabilidad							x	x	2
Estandarización de comunicaciones	x								1
Estandarización de los datos	x								1
Estética			x						1
Estructurado		x		x					2
Evolución					x				1
Exactitud	x	x	x				x	x	5
Expansibilidad	x								1
Explicitud								x	1
Extensibilidad			x			x			2
Facilidad de aprendizaje							x	x	2
Facilidad de auditoría	x								1
Facilidad de cambio							x	x	2
Factores humano			x						1
Flexibilidad	x					x			2

MODELO/ESTÁNDAR vs ATRIBUTOS	McCall	Boehm	FURPS	Dromey	SQAE	Bansiya	ISO 9126	Quint 2	Frecuencia atributo
Frecuencia y severidad de fallos			x						1
Funcionalidad			x	x		x	x	x	5
Generalidad	x		x						2
Genérico				x					1
Herencia						x			1
Holgadamente acoplado				x					1
Homogéneo				x					1
Idoneidad							x	x	2
Independencia					x				1
Independencia de dispositivo		x							1
Independencia de maquina	x								1
Independencia de sistema software	x								1
Ingeniería Humana		x							1
Inicializado				x					1
Instalabilidad			x				x	x	3
Instrumentación	x								1
Integridad	x								1
Interoperabilidad	x						x	x	3
Jerarquía						x			1
Legibilidad		x							1
Localizabilidad (internalización)			x						1
Madurez							x	x	2
Manejabilidad								x	1
Mantenibilidad	x	x	x	x	x		x	x	7
Mensajes						x			1
Modificabilidad		x							1
Modularidad	x				x				2
No redundante				x					1
Operatividad	x						x	x	3
Parametrizado				x					1
Personalización								x	1
Polimorfismo						x			1
Portabilidad	x	x		x	x		x	x	6
Preciso				x					1
Predictibilidad			x						1
Progresivo				x					1
Rango independiente				x					1
Recuperabilidad			x				x	x	3
Remplazabilidad							x	x	2
Rendimiento			x						1
Resuelto				x					1
Reusabilidad	x			x		x		x	4
Seguridad			x				x	x	3
Servicio			x						1

MODELO/ESTÁNDAR vs ATRIBUTOS	McCall	Boehm	FURPS	Dromey	SQAE	Bansiya	ISO 9126	Quint 2	Frecuencia atributo
Simplicidad	x								1
Soportabilidad			x						1
Tamaño de diseño						x			1
Testabilidad	x	x	x				x	x	5
Tiempo de respuesta			x						1
Tiempo medio entre fallos			x						1
Tolerancia a fallos									0
Tolerancia al error	x						x	x	3
Trazabilidad	x							x	2
Usabilidad	x		x	x			x	x	5
Utilización de recursos							x		1
Utilizado				x					1
Variante				x					1
Velocidad			x						1
Total Atributos por modelo	34	20	33	35	11	17	33	36	

Tabla 28. Atributo vs. Modelo de calidad para producto software

El modelo que más atributos posee es Quint2 con 38 atributos y el que menos posee es SQAE con tan solo 11 atributos, entre internos y externos. Los resultados obtenidos acerca de los atributos internos y externos están plasmados en la siguiente tabla (Tabla 29):

MODELO/ESTÁNDAR	Numero de Atributos		
	Internos	Externos	Total
McCall	23	11	34
Boehm	12	8	20
FURPS	27	5	32
Dromey	26	7	33
SQAE	7	4	11
Bansiya	11	6	17
ISO 9126	27	6	33
QUINT2	31	6	37

Tabla 29. Número de atributos por Modelo

El atributo que más aparece en los modelos es Mantenibilidad. Y existen muchos, principalmente internos que solo aparecen en un solo modelo.

ANEXO C: APLICACIÓN DE LOS CRITERIOS A LOS ATRIBUTOS DE CALIDAD ENCONTRADOS

Éste anexo presenta la aplicación de los criterios de clasificación presentados en el capítulo 2 para atributos y la clasificación según espectro.

Criterios:

- **A1: Relación con métricas:** indica si el atributo tiene métricas relacionadas, puede ser por medio directo o indirecto.
 - 1: No fueron encontradas métricas
 - 2: Algunas métricas fueron encontradas pero están incompletas o no son claras.
 - 3: Tiene métricas asociadas claras.

- **A2: Completitud:** refiere a la información encontrada del atributo.
 - 1: No fue encontrada información, es casi nula.
 - 2: La información encontrada es limitada e incompleta.
 - 3: El atributo es suficientemente documentado. Es mencionado por diversos autores.

- **A3: Simplicidad:** Grado en que el atributo es claro y fácil de entender.
 - 1: El atributo no tiene información que sustente su comprensión.
 - 2: El atributo posee información pero es ambigua, puede tener diferentes interpretaciones.
 - 3: El atributo es claro y fácil de entender.

Núm.	ATRIBUTOS	A1	A2	A3	Puntuación
1	Abstracción	3	2	3	8
2	Abstracto	3	2	3	8
3	Accesibilidad	3	3	3	9
4	Acoplamiento	3	2	3	8
5	Adaptabilidad	3	3	3	9
6	Ajustable	1	2	2	5
7	Amigabilidad	3	2	3	8
8	Analizabilidad	3	3	2	8
9	Asignado	1	2	2	5
10	Atractivo	3	3	3	9
11	Auto descripción	2	3	3	8
12	Ayuda	2	2	3	7
13	Capacidad de crecimiento	2	2	2	6
14	Capacidades	1	2	1	4
15	Características	1	2	1	4
16	Claridad	2	2	3	7
17	Co existencia	2	2	2	6
18	Cohesión	3	3	2	8

Núm.	ATRIBUTOS	A1	A2	A3	Puntuación
19	Cohesivo	1	2	2	5
20	Compatibilidad	1	2	2	5
21	Complejidad	3	3	2	8
22	Complejidad	3	3	2	8
23	Comportamiento en el tiempo	2	3	3	8
24	Composición	3	2	3	8
25	Computable	1	2	2	5
26	Comunicatividad	3	2	3	8
27	Concisión	3	2	3	8
28	Confiabilidad	3	3	3	9
29	Configurabilidad	1	2	1	4
30	Conformidad	2	3	3	8
31	Conformidad con la Confiabilidad	2	2	2	6
32	Conformidad con la Eficiencia	2	2	2	6
33	Conformidad con la Funcionalidad	2	2	2	6
34	Conformidad con la Mantenibilidad	2	2	2	6
35	Conformidad con la Usabilidad	2	2	2	6
36	Conformidad con Portabilidad	2	2	2	6
37	Consistencia	3	3	3	9
38	Consistente	3	3	3	9
39	Consumo de recursos	2	3	3	8
40	Control de acceso	2	2	1	5
41	Control de anomalías	1	2	1	4
42	Corrección	3	3	2	8
43	Degradabilidad	2	2	2	6
44	Descripción	1	2	2	5
45	Desempeño	3	2	3	8
46	Directo	1	2	1	4
47	Diseño simple	1	2	1	4
48	Disponibilidad	3	2	3	8
49	Documentación	2	3	3	8
50	Documentado	1	2	2	5
51	Efectividad	1	2	2	5
52	Efectivo	1	2	2	5
53	Eficiencia	3	3	3	9
54	Eficiencia de almacenamiento	2	2	2	6
55	Eficiencia de dispositivo	2	2	2	6
56	Eficiencia de ejecución	2	2	2	6
57	Encapsulamiento	3	3	2	8
58	Encapsulado	2	1	2	5
59	Comprensibilidad	3	3	2	8
60	Entrenamiento	2	2	2	6
61	Especificado	1	2	2	5
62	Estabilidad	2	3	3	8
63	Estandarización de comunicaciones	2	2	2	6

Núm.	ATRIBUTOS	A1	A2	A3	Puntuación
64	Estandarización de los datos	2	2	2	6
65	Estética	1	2	1	4
66	Estructurado	1	2	2	5
67	Evolución	1	2	1	4
68	Exactitud	3	3	2	8
69	Expansibilidad	2	3	2	7
70	Explicitud	2	2	2	6
71	Extensibilidad	1	2	2	5
72	Facilidad de aprendizaje	3	2	3	8
73	Facilidad de auditoria	2	2	2	6
74	Facilidad de cambio	3	3	2	8
75	Factores humano	1	2	1	4
76	Flexibilidad	3	3	3	9
77	Frecuencia y severidad de fallos	1	1	1	3
78	Funcionalidad	3	3	3	9
79	Generalidad	2	3	2	7
80	Genérico	1	2	2	5
81	Herencia	3	2	3	8
82	Holgadamente acoplado	1	2	1	4
83	Homogéneo	1	2	1	4
84	Idoneidad	3	2	3	8
85	Independencia	1	2	1	4
86	Independencia de dispositivo	2	2	2	6
87	Independencia de maquina	2	3	3	8
88	Independencia de sistema software	2	3	3	8
89	Ingeniería Humana	2	3	2	7
90	Inicializado	1	2	1	4
91	instalabilidad	3	2	3	8
92	Instrumentación	2	3	2	7
93	Integridad	1	3	2	6
94	Interoperabilidad	3	3	2	8
95	Jerarquía	3	2	3	8
96	Legibilidad	2	3	2	7
97	Localizabilidad (Internalización)	1	3	2	6
98	Madurez	2	3	2	7
99	Manejabilidad	2	2	2	6
100	Mantenibilidad	3	3	3	9
101	Mensajes	1	2	2	5
102	Modificabilidad	2	2	2	6
103	Modularidad	2	3	3	8
104	No redundante	1	2	2	5
105	Operatividad	3	3	2	8
106	Parametrizado	1	2	2	5
107	Personalización	2	3	3	8
108	Polimorfismo	3	2	3	8

Núm.	ATRIBUTOS	A1	A2	A3	Puntuación
109	Portabilidad	3	3	3	9
110	Preciso	2	2	2	6
111	Predictibilidad	2	2	2	6
112	Progresivo	1	2	2	5
113	Rango independiente	1	2	1	4
114	Recuperabilidad	3	2	3	8
115	Remplazabilidad	3	2	2	7
116	Rendimiento	1	1	1	3
117	Resuelto	1	2	2	5
118	Reusabilidad	3	3	3	9
119	Seguridad	3	3	2	8
120	Capacidad de servicio	1	2	2	5
121	Simplicidad	2	3	2	7
122	Soportabilidad	3	3	3	9
123	Tamaño de diseño	3	3	2	8
124	Testeabilidad	3	3	2	8
125	Tiempo de respuesta	2	2	2	6
126	Tiempo medio entre fallos	2	2	2	6
127	Tolerancia a fallos	2	3	2	7
128	Tolerancia al error	3	3	2	8
129	Trazabilidad	2	3	2	7
130	Usabilidad	3	3	3	9
131	Utilización de recursos	2	2	2	6
132	Utilizado	1	2	1	4
133	Variante	1	2	1	4
134	Velocidad	1	2	2	5

Tabla 30. Puntuación de atributos

Después de revisar los modelos, es posible observar que aparecen 134 atributos, de los cuales 67 son de espectro oscuro, 12 de gris y 55 de claro (ver Tabla 30). La organización de los atributos puede verse en el Capítulo 3.

ANEXO D: APLICACIÓN DE LOS CRITERIOS A LAS MÉTRICAS DE CALIDAD ENCONTRADAS

Éste anexo muestra la aplicación de los criterios de clasificación para métricas presentados en el capítulo 2 (Ver tabla 31).

Criterios:

- **Mt1: *Compleitud*:** Referido a la bibliografía, la cantidad de documentos encontrados donde aparece la métrica, y qué tan documentada esta.
 - 1: No existe información sobre la métrica o no es suficiente. La mención por parte de los autores es casi nula.
 - 2: Existe información pero está incompleta, es posible hacer interpretaciones sobre la métrica. La mención hecha por autores es limitada.
 - 3: la métrica está bien documentada. Es mencionada por diversos autores.

- **Mt2: *Computable*:** Indica si es posible llevar la métrica a términos matemáticos para posteriormente ser puesta en funcionamiento a través de un programa computacional.
 - 1: La métrica es totalmente subjetiva; no es posible computarizarla
 - 2: Es posible computarizarla. Tal vez necesita intervención humana para ser calculada.
 - 3: Es posible computarizar la métrica, está en términos matemáticos claros.

- **Mt3: *Objetividad*:** Grado en que la métrica evita interpretaciones individuales. En cualquier opinión los valores han de ser idénticos
 - 1 y 2: La métrica puede tener interpretaciones diferentes.
 - 3: La métrica no tiene interpretaciones diferentes.

- **Mt4: *Simple*:** Grado en que la métrica es fácil entender por el lector. Tanto su definición como su uso es simple. Una métrica simple no puede ser ambigua.
 - 1: La métrica es compleja o su interpretación es ambigua.
 - 2: La métrica es compleja pero puede ser entendida con algo de estudio.
 - 3: La métrica es fácil de entender y usar.

Núm.	Métrica	Criterios				Total
		Mt1	Mt2	Mt3	Mt4	
1	Medida de integridad	1	2	1	3	7
2	Eficiencia esencial	1	2	2	2	7
3	Concordancia de tarea	1	2	1	2	6
4	Visibilidad de tarea	1	1	1	1	4
5	Uniformidad de distribución	1	2	1	1	5
6	CoHerencia visual	1	2	1	1	5
7	Disponibilidad de construcción en pruebas de	2	2	1	3	8

Núm.	Métrica	Criterios				Total
		Mt1	Mt2	Mt3	Mt4	
	función					
8	Eficiencia de re – prueba	2	2	2	3	9
9	Prueba de reactivación	2	1	1	3	7
10	Compleitud de construcción en prueba de función	2	1	2	2	7
11	Autonomía de capacidad de prueba	2	2	1	1	6
12	Observación del progreso de prueba	2	2	2	2	8
13	InterCambiabilidad de datos (Basado en formato de datos)- externa	2	1	1	2	6
14	InterCambiabilidad de datos (Basado en el intento de éxito del usuario)	2	1	1	2	6
15	InterCambiabilidad de datos(Basado en formato de datos)- interna	2	1	1	2	6
16	CoHerencia de interfaz (Protocolo)	2	1	1	2	6
17	Número de hijos o descendientes	3	3	3	3	12
18	Beneficio de Reutilización	1	2	1	1	5
19	Aprovechamiento de Reutilización	1	2	1	1	5
20	Acoplamiento entre objetos	3	3	3	2	11
21	Carencia de cohesión en los métodos	3	2	2	2	9
22	Número promedio de antecesores	2	1	3	3	9
23	Abstracción	2	2	2	2	8
24	Lista de chequeo de Auditoria de acceso	1	2	1	2	6
25	Clase de acoplamiento directo	2	1	3	3	9
26	Factor de acoplamiento	3	3	3	2	11
27	Acoplamiento de otros módulos	1	2	2	3	8
28	Acoplamiento de Abstracción de datos	3	2	3	1	9
29	Acoplamiento de paso de masajes	3	2	1	1	7
30	Número de clases llamadas	1	2	1	1	5
31	Número de llamadas de operación	1	2	1	1	5
32	Adaptabilidad de estructuras de datos-externo	2	2	1	1	6
33	Adaptabilidad de entorno hardware-externo	2	2	1	2	7
34	Adaptabilidad de entorno organizacional -externo	2	2	1	2	7
35	Amigabilidad de usuario de puerto -externo	2	2	1	2	7
36	Adaptabilidad de entorno de sistema software – externo	2	2	1	2	7
37	Adaptabilidad de estructuras de datos-interno	2	2	1	2	7
38	Adaptabilidad de entorno hardware-interno	2	2	1	2	7
39	Adaptabilidad de entorno organizacional-interno	2	2	1	2	7
40	Amigabilidad de usuario de puerto-interno	2	2	1	2	7
41	Adaptabilidad de entorno de sistema software – interno	2	2	1	2	7
42	Cohesión entre los métodos en la clase	3	NA	2	2	7
43	Medida de agregación	2	2	3	3	10
44	Número de atributos	2	2	3	3	10
45	Compleitud 1	1	NA	1	1	3

Núm.	Métrica	Criterios				Total
		Mt1	Mt2	Mt3	Mt4	
46	Compleitud 2	1	NA	1	1	3
47	Compleitud 3	1	NA	1	1	3
48	Compleitud 4	1	NA	1	1	3
49	Compleitud 5	1	NA	1	1	3
50	Compleitud 6	1	NA	1	1	3
51	Compleitud 7	1	NA	1	1	3
52	Compleitud 8	1	NA	1	1	3
53	Compleitud 9	1	NA	1	1	3
54	Compleitud 10	1	NA	1	1	3
55	Compleitud 11	1	NA	1	1	3
56	Compleitud 12	1	NA	1	1	3
57	Compleitud 13	1	NA	1	1	3
58	Compleitud 14	1	NA	1	1	3
59	Compleitud 15	1	NA	1	1	3
60	Requisitos de completitud (Funcionales)	1	2	2	2	7
61	Requisitos de completitud (No Funcionales)	1	2	2	2	7
62	Lista de chequeo de completitud	1	2	1	2	6
63	Compleitud de descripción- Externo	2	2	1	2	7
64	Accesibilidad de demostración	2	2	1	2	7
65	Accesibilidad de demostración en uso	2	2	1	2	7
66	Efectividad de demostración	2	2	1	2	7
67	Funciones evidentes- Externo	2	2	1	2	7
68	Comprensibilidad de función – Externo	2	2	1	2	7
69	Entrada y salida entendible	2	2	1	2	7
70	Compleitud de descripción- Interno	2	2	1	2	7
71	Capacidad de demostración	2	2	1	2	7
72	Funciones evidentes – Interno	2	2	1	2	7
73	Comprensibilidad de función – Interno	2	2	1	2	7
74	Concisión 1	1	NA	1	1	3
75	Concisión 2	1	NA	1	1	3
76	Concisión 3	1	NA	1	1	3
77	Concisión 4	1	NA	1	1	3
78	Concisión 5	1	NA	1	1	3
79	Concisión 6	1	NA	1	1	3
80	Concisión 7	1	NA	1	1	3
81	Concisión 8	1	NA	1	1	3
82	Métricas de Hastead	2	2	2	2	8
83	Conformidad funcional- Externo	2	2	2	2	8
84	Conformidad con interface estándar	2	2	2	2	8
85	Conformidad funcional – interno	2	2	2	2	8
86	Conformidad estándar inter sistema	2	2	2	2	8
87	Conformidad con la Confiabilidad- Externo	2	2	2	2	8
88	Conformidad con la Confiabilidad- interno	2	2	2	2	8
89	Conformidad con la Usabilidad - externo	2	2	2	2	8

Núm.	Métrica	Criterios				Total
		Mt1	Mt2	Mt3	Mt4	
90	Conformidad con la Usabilidad – interno	2	2	2	2	8
91	Conformidad con la Eficiencia – Externo	2	2	2	2	8
92	Conformidad con la Eficiencia - Interno	2	2	2	2	8
93	Conformidad con la Mantenibilidad – Externo	2	2	2	2	8
94	Conformidad con la Mantenibilidad – Interno	2	2	2	2	8
95	Conformidad con la Portabilidad- Externo	2	2	2	2	8
96	Conformidad con la Portabilidad – Interno	2	2	2	2	8
97	Consistencia 1	1	NA	1	1	3
98	Consistencia 2	1	NA	1	1	3
99	Consistencia 3	1	NA	1	1	3
100	Consistencia 4	1	NA	1	1	3
101	Consistencia 5	1	NA	1	1	3
102	Consistencia 6	1	NA	1	1	3
103	Consistencia 7	1	NA	1	1	3
104	Consistencia 8	1	NA	1	1	3
105	Consistencia 9	1	NA	1	1	3
106	Consistencia 10	1	NA	1	1	3
107	Consistencia 11	1	NA	1	1	3
108	Consistencia 12	1	NA	1	1	3
109	CoHerencia de procedimiento	1	2	1	2	6
110	Consistencia de datos	1	2	1	2	6
111	Métrica de acceso de datos	2	2	3	3	10
112	Porción de métodos ocultos	3	3	3	2	11
113	Porción de atributos ocultos	3	3	3	2	11
114	Carencia de ambigüedad	1	2	1	1	5
115	Exactitud 1	1	NA	1	1	3
116	Exactitud de expectativa	2	2	2	2	8
117	Exactitud computacional- externo	2	2	2	2	8
118	Precisión - externo	2	2	2	2	8
119	Exactitud computacional- interno	2	2	2	2	8
120	Precisión- interno	2	2	2	2	8
121	Lista de chequeo de precisión	1	2	1	2	6
122	Medida de expansión de almacenamiento de los datos	1	2	1	2	6
123	Medida de Extensibilidad	1	2	1	2	6
124	Grado en que un modulo es referenciado por otros módulos	1	2	1	2	6
125	Implementación para la lista de chequeo de generalidad	1	2	1	2	6
126	Medida de Abstracción funcional	2	2	1	1	6
127	Porción de métodos heredados	3	3	3	2	11
128	Porción de atributos heredados	3	3	3	2	11
129	Índice de especialización por clase	2	3	1	1	7
130	Árbol de profundidad de Herencia	3	3	3	3	12

Núm.	Métrica	Criterios				Total
		Mt1	Mt2	Mt3	Mt4	
131	Número de métodos heredados	1	2	3	1	7
132	Número de método remplazados	1	2	2	1	6
133	Número de métodos añadidos	1	2	3	2	8
134	Número de nuevos métodos	1	1	3	2	7
135	Número de descendientes	1	2	3	2	8
136	Número promedio de clases derivadas	1	2	2	2	7
137	Promedio de altura de la jerarquía	1	2	2	2	7
138	Independencia de dispositivo 1	1	NA	1	2	4
139	Independencia de dispositivo 2	1	NA	1	1	3
140	Independencia de dispositivo 3	1	NA	1	1	3
141	Independencia de dispositivo 4	1	NA	1	1	3
142	Independencia de dispositivo 5	1	NA	1	1	3
143	Independencia de dispositivo 6	1	NA	1	1	3
144	Medida de independencia de maquina	1	2	1	2	6
145	Medida de independencia de sistema SW	1	2	1	2	6
146	Fácil de instalar	2	2	2	2	8
147	Facilidad de re instalación - Externo	2	2	2	2	8
148	Facilidad de re instalación - interno	2	2	2	2	8
149	Esfuerzo de instalación	2	2	2	2	8
150	Flexibilidad de instalación	2	2	2	2	8
151	Medida de prueba de modulo	1	2	1	2	6
152	Medida de prueba de integración	1	2	1	2	6
153	Medida de prueba de sistema	1	2	1	2	6
154	Medida de Estabilidad	1	2	1	2	6
155	Medida de implementación modular	1	2	1	2	6
156	Consistencia operacional en uso	2	2	2	2	8
157	Corrección del error	2	2	2	2	8
158	Corrección del error en uso	2	2	2	2	8
159	Disponibilidad del valor por defecto en uso	2	2	2	2	8
160	Comprensibilidad de mensaje en uso	2	2	2	2	8
161	Auto explicación de mensajes de error	2	2	2	2	8
162	Recuperabilidad de error operacional en uso	2	2	2	2	8
163	Tiempo entre operaciones de error humano en uso	2	2	2	2	8
164	Capacidad de deshacer	2	2	2	2	8
165	Personalización – externo	2	2	2	2	8
166	Operación de reducción de procedimiento	2	2	2	2	8
167	Accesibilidad física – externo	2	2	2	2	8
168	Lista de chequeo de validación de entrada	1	2	2	2	7
169	Operación de usuario, capacidad de cancelar	3	2	2	2	9
170	Operación de usuario, deshacer	3	3	2	2	10
171	Personalización – interno	2	2	2	2	8
172	Accesibilidad física – interno	2	2	2	2	8
173	Estado de funcionamiento de la capacidad de	2	2	2	2	8

Núm.	Métrica	Criterios				Total
		Mt1	Mt2	Mt3	Mt4	
	vigilancia					
174	Consistencia operacional	2	2	2	2	8
175	Claridad de mensaje	2	2	2	2	8
176	Claridad de elemento de interface	2	2	2	2	8
177	Recuperabilidad de error operacional	2	2	2	2	8
178	Lista de chequeo de operatividad	1	2	1	2	6
179	Número de métodos polimórficos	3	2	2	2	9
180	Factor polimórfico	3	2	3	2	10
181	Capacidad de auditoría de acceso – externo	2	2	2	2	8
182	Capacidad de control de acceso - externo	2	2	2	2	8
183	Prevención de corrupción de datos – externo	2	2	2	2	8
184	Capacidad de auditoría de acceso – interno	2	2	2	2	8
185	Capacidad de control de acceso – interno	2	2	2	2	8
186	Prevención de corrupción de datos – interno	2	2	2	2	8
187	Encriptación de datos	2	2	2	2	8
188	Medida de estructura de diseño	1	2	1	2	6
189	Uso de lenguaje Estructurado o procesador	1	2	1	2	6
190	Medida de Complejidad	1	2	1	2	6
191	Medida de simplicidad de técnicas de codificación	1	2	1	2	6
192	Referencias cruzadas relacionando módulos a requerimientos	1	2	1	2	6
193	Utilización de dispositivos de E/S	2	2	2	2	8
194	Limites de carga de E/S	2	2	2	2	8
195	Cociente de cumplimiento medio de E/S	2	2	2	2	8
196	Tiempo de espera de usuario para utilización de dispositivos de E/S	2	2	2	2	8
197	Utilización máxima de memoria	2	2	2	2	8
198	Ocurrencia media de errores de memoria	2	2	2	2	8
199	Cociente error/tiempo de memoria	2	2	2	2	8
200	Utilización máxima de transmisión	2	2	2	2	8
201	Balance de utilización del dispositivo de comunicación	2	2	2	2	8
202	Ocurrencia media de error de transmisión	2	2	2	2	8
203	Media de error de transmisión por vez	2	2	2	2	8
204	Utilización de capacidad de transmisión	2	2	2	2	8
205	Utilización de E/S	2	2	2	2	8
206	Densidad de utilización de mensajes E/S	2	2	2	2	8
207	Utilización de memoria	2	2	2	2	8
208	Densidad de utilización de mensajes de memoria	2	2	2	2	8
209	Utilización de transmisión	2	2	2	2	8
210	Comunicatividad 1	1	NA	1	1	3
211	Comunicatividad 2	1	NA	1	1	3
212	Comunicatividad 3	1	NA	1	1	3
213	Comunicatividad 4	1	NA	1	1	3

Núm.	Métrica	Criterios				Total
		Mt1	Mt2	Mt3	Mt4	
214	Comunicatividad 5	1	NA	1	1	3
215	Comunicatividad 6	1	NA	1	1	3
216	Comunicatividad 7	1	NA	1	1	3
217	Comunicatividad 8	1	NA	1	1	3
218	Comunicatividad 9	1	NA	1	1	3
219	Comunicatividad 10	1	NA	1	1	3
220	Comunicatividad 11	1	NA	1	1	3
221	Medida de interface de entrada de usuario	1	2	1	2	6
222	Medida de interface de salida de usuario	1	2	1	2	6
223	Interacción atractiva – externo	2	2	2	2	8
224	Personalización de apariencia de interfaz- externo	2	2	2	2	8
225	Interacción atractiva – interno	2	2	2	2	8
226	Personalización de apariencia de interfaz –interno	2	2	2	2	8
227	Lista de chequeo de entrenamiento	1	2	1	2	6
228	Fácil aprendizaje de función	2	2	2	2	8
229	Fácil aprendizaje de realizar una tarea en uso	2	2	2	2	8
230	Efectividad de la documentación de usuario y sistema de ayuda	2	2	2	2	8
231	Efectividad de la documentación de usuario y/o sistema de ayuda en uso	2	2	2	2	8
232	Accesibilidad a ayuda	2	2	2	2	8
233	Frecuencia a ayuda	2	2	2	2	8
234	Completitud de documentación de usuario y/o Facilidad de ayuda	2	2	2	2	8
235	Adecuación funcional - externo	2	2	2	2	8
236	Completitud de la implementación funcional – externo	2	2	2	2	8
237	Cobertura de la implementación funcional – externo	2	2	2	2	8
238	Estabilidad (volatilidad) de la especificación funcional – externo	2	2	2	2	8
239	Adecuación funcional – interno	2	2	2	2	8
240	Completitud de la implementación funcional – interno	2	2	2	2	8
241	Cobertura de la implementación funcional – interno	2	2	2	2	8
242	Estabilidad (volatilidad) de la especificación funcional – interno	2	2	2	2	8
243	Legibilidad 1	1	NA	1	1	3
244	Legibilidad 2	1	NA	1	1	3
245	Legibilidad 3	1	NA	1	1	3
246	Legibilidad 4	1	NA	1	1	3
247	Legibilidad 5	1	NA	1	1	3
248	Legibilidad 6	1	NA	1	1	3

Núm.	Métrica	Criterios				Total
		Mt1	Mt2	Mt3	Mt4	
249	Legibilidad 7	1	NA	1	1	3
250	Legibilidad 8	1	NA	1	1	3
251	Legibilidad 9	1	NA	1	1	3
252	Evasión de fracaso	2	2	2	2	8
253	Evasión de falla -externo	2	2	2	2	8
254	Evasión de operación incorrecta –externo	2	2	2	2	8
255	Evasión de falla - interno	2	2	2	2	8
256	Evasión de operación incorrecta – interno	2	2	2	2	8
257	Lista de chequeo de control de tolerancia a error	1	2	1	2	6
258	Recuperación desde la lista de chequeo de datos de entrada impropios	1	2	1	2	6
259	Recuperación desde la lista de chequeo de fallas computacionales	1	2	1	2	6
260	Recuperación desde la lista de chequeo de errores desde HW	1	2	1	2	6
261	Recuperación desde la lista de chequeo de errores de dispositivo	1	2	1	2	6
262	Complejidad estructural	2	3	2	2	9
263	Complejidad de datos	2	3	2	2	9
264	Complejidad de sistema	2	3	2	2	9
265	Métrica de diseño arquitectónico	1	2	2	1	6
266	Respuesta para una clase	3	3	2	1	9
267	Métodos ponderados por clase	3	3	2	2	10
268	Complejidad ciclomatica media	1	3	2	2	8
269	Complejidad ciclomatica	3	2	2	2	9
270	Máxima Complejidad ciclomatica	1	2	2	2	7
271	Extensión a Complejidad ciclomatica	1	2	2	1	6
272	Métrica de Complejidad ciclomatica extendida	1	2	2	1	6
273	Número de métodos (locales)	3	3	3	2	11
274	Número de clases	1	3	3	2	9
275	Número de paquetes	1	NA	3	2	6
276	Flujo de información de Complejidad	1	NA	1	1	3
277	Uso continuo de datos -externo	2	2	2	2	8
278	Inclusión de funciones -externo	2	2	2	2	8
279	Consistencia funcional de soporte de usuario	2	2	2	2	8
280	Uso continuo de datos -interno	2	2	2	2	8
281	Inclusión de funciones - interno	2	2	2	2	8
282	Tamaño de clase interfaz	3	2	3	1	9
283	Tamaño de diseño en clases	2	2	3	2	9
284	Size 1	3	2	3	2	10
285	Size 2	3	2	3	2	10
286	Accesibilidad 1	1	NA	1	1	3
287	Accesibilidad 2	1	NA	1	1	3
288	Accesibilidad 3	1	NA	1	1	3

Núm.	Métrica	Criterios				Total
		Mt1	Mt2	Mt3	Mt4	
289	Accesibilidad 4	1	NA	1	1	3
290	Accesibilidad 5	1	NA	1	1	3
291	Estructuración 1	1	NA	1	1	3
292	Estructuración 2	1	NA	1	1	3
293	Estructuración 3	1	NA	1	1	3
294	Estructuración 4	1	NA	1	1	3
295	Estructuración 5	1	NA	1	1	3
296	Estructuración 6	1	NA	1	1	3
297	Estructuración 7	1	NA	1	1	3
298	Estructuración 8	1	NA	1	1	3
299	Estructuración 9	1	NA	1	1	3
300	Auto descripción 1	1	NA	1	1	3
301	Auto descripción 2	1	NA	1	1	3
302	Auto descripción 3	1	NA	1	1	3
303	Auto descripción 4	1	NA	1	1	3
304	Auto descripción 5	1	NA	1	1	3
305	Auto descripción 6	1	NA	1	1	3
306	Auto descripción 7	1	NA	1	1	3
307	Auto descripción 8	1	NA	1	1	3
308	Auto descripción 9	1	NA	1	1	3
309	Auto descripción 10	1	NA	1	1	3
310	Auto descripción 11	1	NA	1	1	3
311	Cantidad de comentarios	1	2	1	2	6
312	Efectividad de la medida de comentarios	1	2	1	2	6
313	Descripción de la medida de lenguaje de implementación	1	2	1	2	6
314	Requisitos de Desempeño asignados a diseño	1	2	1	2	6
315	Proceso iterativo de medida de Eficiencia	1	2	1	2	6
316	Medida de Eficiencia de uso de datos	1	2	1	2	6
317	Eficiencia de dispositivo 1	1	NA	1	1	3
318	Eficiencia de dispositivo 2	1	NA	1	1	3
319	Eficiencia de dispositivo 3	1	NA	1	1	3
320	Eficiencia de dispositivo 4	1	NA	1	1	3
321	Lista de chequeo de comunicaciones común	1	2	1	2	6
322	Lista de chequeo de datos común	1	2	1	2	6
323	Facilidad de crecimiento 1	1	NA	1	1	3
324	Disponibilidad	2	2	1	2	7
325	Tiempo medio de inactividad	2	2	1	2	7
326	Reiniciabilidad	2	2	1	2	7
327	Restaurabilidad -externo	2	2	1	2	7
328	Efectividad de restauración -externo	2	2	1	2	7
329	Restaurabilidad -interno	2	2	1	2	7
330	Efectividad de restauración -interno	2	2	1	2	7
331	Tiempo medio entre fallos	2	2	3	2	9

Núm.	Métrica	Criterios				Total
		Mt1	Mt2	Mt3	Mt4	
332	Número de jerarquías	3	2	2	2	9
333	Densidad de falla latente estimada	2	2	2	2	8
334	Densidad de falla contra los casos de prueba	2	2	2	2	8
335	Resolución de fallas	2	2	2	2	8
336	Densidad de fallas	2	2	2	2	8
337	Remoción de fallas- externo	2	2	2	2	8
338	Cobertura de prueba	2	2	2	2	8
339	Madurez de prueba	2	2	2	2	8
340	Detección de falla	2	2	2	2	8
341	Remoción de falla- interno	2	2	2	2	8
342	Adecuación de prueba	2	2	2	2	8
343	Índice de madurez del software	1	2	2	1	6
344	Tiempo de respuesta - externo	2	2	2	2	8
345	Tiempo de respuesta (tiempo medio de respuesta)	2	2	2	2	8
346	Tiempo de respuesta (peor caso en el cociente de tiempo de respuesta)	2	2	2	2	8
347	Rendimiento	2	2	2	2	8
348	Rendimiento (Cantidad media de rendimiento)	2	2	2	2	8
349	Rendimiento (peor caso en el cociente de rendimiento)	2	2	2	2	8
350	Tiempo de espera - externo	2	2	2	2	8
351	Tiempo de espera (tiempo medio de espera)	2	2	2	2	8
352	Tiempo de espera (peor caso del cociente de tiempo de espera)	2	2	2	2	8
353	Tiempo esperando	2	2	2	2	8
354	Tiempo de respuesta - interno	2	2	2	2	8
355	Tiempo de rendimiento – interno	2	2	2	2	8
356	Tiempo de espera – interno	2	2	2	2	8
357	Capacidad de rastro de auditoria	2	2	2	2	8
358	Soporte de la función de diagnostico	2	2	2	2	8
359	Capacidad de análisis de falla	2	2	2	2	8
360	Eficiencia de análisis de falla	2	2	2	2	8
361	Capacidad de monitorear estado	2	2	2	2	8
362	Registro de actividad	2	2	2	2	8
363	Preparación de la función de diagnostico	2	2	2	2	8
364	Violación de Complejidad	1	2	1	2	6
365	Porcentaje de éxito del cambio	2	2	2	2	8
366	Modificaciones a la localización de impacto – externo	2	2	2	2	8
367	Impacto del cambio	2	2	2	2	8
368	Modificaciones a la localización de impacto – interno	2	2	2	2	8
369	Variable global y uso de temporizador	1	2	1	1	5

Núm.	Métrica	Criterios				Total
		Mt1	Mt2	Mt3	Mt4	
370	Reasignación de parámetro	1	2	3	1	7
371	Co existencia disponible - externo	2	2	2	2	8
372	Co existencia disponible - interno	2	2	2	2	8
373	Eficiencia del ciclo de cambio	2	2	2	2	8
374	Tiempo trascurrido de implementación de cambio	2	2	2	2	8
375	Modificación de la Complejidad	2	2	2	2	8
376	Modificabilidad de parámetros	2	2	2	2	8
377	Capacidad de control de cambios del software	2	2	2	2	8
378	Capacidad de registro de cambios	2	2	2	2	8
379	Duplicación de código	1	2	1	1	5
380	Número máximo de violación de referencias	1	2	1	1	5
381	Punto de función	3	3	2	2	10
382	Métrica del bang	2	2	3	1	8
383	Número de elementos	1	2	3	3	9
384	Líneas de código por método	3	3	3	2	11
385	Métrica del tamaño	1	3	3	2	9
386	Métrica de profundidad	1	3	3	2	9
387	Métrica de anchura	1	2	3	2	8
388	Métrica de relación arco - nodo	1	2	3	2	8
389	Número de métodos públicos	1	2	3	2	8
390	Número de variables publicas por clase	1	2	2	2	7
391	Número de variables por clase	1	2	2	2	7
392	Número de métodos de clase	1	2	2	2	7
393	Número de métodos instancia públicos	1	2	1	2	6
394	Número de métodos instancia	1	2	1	2	6
395	Número de variables instancia	1	2	1	2	6
396	Número de mensajes enviados	1	2	2	1	6
397	Tamaño relativo	1	1	2	1	5
398	Índice de matenibilidad	1	2	2	1	6
399	Medida de Eficiencia de almacenamiento	1	2	1	2	6
400	Lista de chequeo de control de acceso	1	2	1	2	6
401	Atributos locales	1	2	3	3	9
402	Operaciones totales	1	2	3	3	9
403	Operaciones locales	1	2	1	2	6
404	Operaciones remplazadas	1	2	1	2	6
405	Operaciones de clase amplia	1	2	1	1	5
406	Atributos de tipos abstractos de datos	1	2	1	1	5
407	Tipos abstractos de datos referenciados	1	2	1	1	5
408	Promedio de parámetros por método	1	2	2	2	7
409	Número de punto y comas (NOSC)	1	2	2	2	7
410	Número de atributos y métodos (NAANM)	1	2	2	2	7
411	Número ponderado de atributos y métodos (WNAAWNMM)	1	2	2	2	7
412	Complejidad de atributo	1	2	2	2	7

Núm.	Métrica	Criterios				Total
		Mt1	Mt2	Mt3	Mt4	
413	Complejidad operacional	1	2	2	2	7
414	Complejidad de argumento de operación	1	2	2	2	7
415	Métrica de cohesión	1	2	2	2	7
416	Número de métodos locales privados	1	2	2	2	7
417	Ponderado de clases abstractas a cociente de atributo (WACAAR)	1	2	2	2	7
418	Ponderado de métodos privados a cociente ponderado de métodos (WPMMR)	1	2	2	2	7
419	Número de estado estables en una clase (NSSC)	1	2	2	2	7
420	Número de eventos distintos en una clase (NDEC)	1	2	2	2	7
421	Tamaño del dominio del mensaje	1	2	2	2	7
422	Número de vagos	1	2	2	2	7
423	Porción de los distintos eventos del Número de métodos de interface	1	2	2	2	7
424	Tamaño del vocabulario del mensaje	1	2	2	2	7
425	Métrica de acoplamiento de operación	1	2	2	2	7
426	Métrica de acoplamiento de clase	1	2	2	2	7
427	Violaciones a la ley de Demeter	1	2	2	2	7
428	Número total de clases	1	2	2	2	7
429	Número de clases de máquina de estado finito	1	2	2	2	7
430	Número de clases de tipo abstracto de datos	1	2	2	2	7
431	Número de máquina de estado finito – interacciones maquina de estado finito	1	2	2	2	7
432	Número de máquina de estado finito – interacciones tipo abstracto de dato	1	2	2	2	7
433	Número de clases agrupadas	1	NA	2	2	5
434	Asociación de Complejidad	1	2	2	2	7
435	Número total de contratos	1	2	2	2	7
436	Cuenta de responsabilidad para una clase	1	2	2	2	7
437	Responsabilidades heredadas por una clase	1	2	2	2	7
438	Cuenta de responsabilidades normalizadas por una clase	1	2	2	2	7
439	Número de tipos colaboradores	1	2	2	2	7
440	Número de colaboradores distintos	1	2	2	2	7
441	Promedio de tamaño de vocabulario colaborador por responsabilidad	1	2	2	2	7
442	Métrica de cohesión de clase	1	2	2	2	7
443	Número de clases cliente	1	2	2	2	7
444	Promedio de cuenta de responsabilidades de clase (ARCC)	1	2	2	2	7
445	Desviación estándar de cuenta de responsabilidades de clases (SDRCC)	1	2	2	2	7
446	Colaboraciones totales (TC)	1	2	2	2	7

Núm.	Métrica	Criterios				Total
		Mt1	Mt2	Mt3	Mt4	
447	Número promedio de distintas colaboraciones de clases (ANDC)	1	2	2	2	7
448	Desviación estándar de Número promedio de distintas colaboraciones de clases (SDNDC)	1	2	2	2	7
449	Número de clases del tipo de máquina de estado finito (NFSM)	1	2	2	2	7
450	Número de clases del tipo: tipo abstracto de datos (NADT)	1	1	2	2	6
451	Número de clases jerárquicas (NCH)	1	2	2	2	7
452	Instalabilidad	1	3	2	2	8
453	Acoplamiento de Abstracción de datos de paquetes	1	2	2	2	7
454	Cambio de dependencia entre las clases	1	2	2	2	7
455	Ajuste de cohesión en la clase	1	2	2	2	7
456	Ajuste de cohesión en el paquete	1	2	2	2	7
457	Perdida de cohesión en la clase	1	2	2	2	7
458	Falta de documentación	1	2	2	2	7
459	Localidad de los datos	1	2	2	2	7
460	Densidad de defectos	1	2	2	2	7
461	Promedio de parámetros del método	1	2	2	2	7
462	Acoplamiento para componentes súper especializados	1	2	1	2	6
463	Tasa de cobertura	1	1	1	2	5
464	Porcentaje de cobertura reducido	1	1	1	2	5
465	Cociente de la especificación funcional de cambio	1	1	1	2	5
466	Relación de solicitud de mejoramiento	1	1	1	2	5
467	Relación de reclamación	1	1	1	2	5
468	Cociente de fracaso	1	1	1	2	5
469	Cociente de dígitos significativos	1	1	1	2	5
470	Relación de Conformidad con el manual	1	1	1	2	5
471	Cociente del tratamiento de redondeo	1	1	1	2	5
472	Esfuerzo por la interacción	1	1	1	2	5
473	Cociente de formato de datos coincidente	1	1	1	2	5
474	Cociente de caracteres coincidentes	1	1	1	2	5
475	Cociente de interfaces coincidentes	1	1	1	2	5
476	Proporción de estándar observado	1	1	1	2	5
477	Proporción de formato de datos estandarizado	1	1	1	2	5
478	Proporción de formato de medio estandarizado	1	1	1	2	5
479	Proporción de carácter estandarizado	1	1	1	2	5
480	Proporción de interface estandarizada	1	1	1	2	5
481	Relación de expresión de Conformidad	1	1	1	2	5
482	Resistencia	1	1	1	2	5
483	Resistencia a Hacker	1	1	1	2	5
484	Relación de cifrado de datos	1	1	1	2	5

Núm.	Métrica	Criterios				Total
		Mt1	Mt2	Mt3	Mt4	
485	Relación de historia de acceso	1	1	1	2	5
486	Relación de daño de datos	1	1	1	2	5
487	Esfuerzo de control de operación	1	1	1	2	5
488	Facilidad de control de operación	1	1	1	2	5
489	Densidad de fallas del producto	1	1	1	2	5
490	Estabilidad del producto	1	1	1	2	5
491	Densidad de prueba	1	1	1	2	5
492	Disturbios	1	1	1	2	5
493	Vulnerabilidad	1	1	1	2	5
494	Valor de integridad	1	1	1	2	5
495	Proporción de detección de error en la operación/entrada	1	1	1	2	5
496	Tiempo medio de reparación	1	1	1	2	5
497	Tiempo medio de interrupción	1	1	1	2	5
498	Tiempo medio de reinicio	1	1	1	2	5
499	Vida media de la falla	1	1	1	2	5
500	Completa relación de la disponibilidad de tiempo	1	1	1	2	5
501	Relación de disponibilidad relativa	1	1	1	2	5
502	Tiempo de retroceso	1	1	1	2	5
503	Tasa de Comprensibilidad	1	1	1	2	5
504	Puntuación de legibilidad	1	1	1	2	5
505	Concepto de claridad	1	1	1	2	5
506	Disponibilidad de demostración de software	1	1	1	2	5
507	Claridad de uso	1	1	1	2	5
508	Disponibilidad de datos de entrada / salida de elementos de la lista	1	1	1	2	5
509	Habilidad de reconocer parámetros modificables	1	1	1	2	5
510	Tiempo promedio de aprendizaje	1	1	1	2	5
511	Usabilidad del manual	1	1	1	2	5
512	Disponibilidad del manual	1	1	1	2	5
513	Disponibilidad de las funciones de pre-aprendizaje	1	1	1	2	5
514	Disponibilidad de las funciones de aprendizaje	1	1	1	2	5
515	Disponibilidad de las funciones de ayuda	1	1	1	2	5
516	Tasa de errores de usuario	1	1	1	2	5
517	Tiempo requerido para aprender una operación	1	1	1	2	5
518	Esfuerzo requerido para aprender una operación	1	1	1	2	5
519	Juicio de experto sobre operatividad	1	1	1	2	5
520	Operatividad comparada con la muestra	1	1	1	2	5
521	Operatividad en la practica	1	1	1	2	5
522	Configuración del tiempo de operación	1	1	1	2	5
523	Puesta en marcha procedimientos de instalación con la interacción humana	1	1	1	2	5
524	Disponibilidad de puesta en marcha de	1	1	1	2	5

Núm.	Métrica	Criterios				Total
		Mt1	Mt2	Mt3	Mt4	
	rendimiento					
525	Facilidad de configurar	1	1	1	2	5
526	Disponibilidad de puesta en marcha de instalación reinicie	1	1	1	2	5
527	Disponibilidad de puesta en marcha de instalación de preparación	1	1	1	2	5
528	Disponibilidad de puesta en marcha de instalación de confirmación	1	1	1	2	5
529	Proporción del valor por defecto la disponibilidad	1	1	1	2	5
530	Uniformidad de comandos	1	1	1	2	5
531	Consistencia en términos del mensaje	1	1	1	2	5
532	Claridad de mensaje	1	1	1	2	5
533	Adaptabilidad al nivel de dificultad	1	1	1	2	5
534	Uniformidad de la pantalla de manipulación de operaciones	1	1	1	2	5
535	Estabilidad de áreas de entrada/salida en pantalla	1	1	1	2	5
536	Numero de pulsaciones del teclado	1	1	1	2	5
537	Disponibilidad de esfuerzo reducido para operaciones repetidas	1	1	1	2	5
538	Tiempo medio entre operaciones de error humano	1	1	1	2	5
539	Hora de apagado de la operación	1	1	1	2	5
540	Proporción de disponibilidad de la función guía	1	1	1	2	5
541	Proporción de habilidad de cancelación de la operación de error humano	1	1	1	2	5
542	Habilidad para enfatizar expresiones	1	1	1	2	5
543	Tiempo de respuesta para el usuario	1	1	1	2	5
544	Tiempo de pantalla	1	1	1	2	5
545	Juicio de expertos sobre la explicitud	1	1	1	2	5
546	Tiempo inseguro	1	1	1	2	5
547	Proporción del reporte de estado	1	1	1	2	5
548	Tiempo inseguro en la practica	1	1	1	2	5
549	Reporte de progreso o estado de la proporción de disponibilidad	1	1	1	2	5
550	Proporción de configurabilidad	1	1	1	2	5
551	Esfuerzo de de configurabilidad	1	1	1	2	5
552	Juicio de usuario sobre el atractivo	1	1	1	2	5
553	Proporción de la función de reconocimiento	1	1	1	2	5
554	Proporción de la función de usuario	1	1	1	2	5
555	Proporción de texto expuesto	1	1	1	2	5
556	Proporción normalizada de texto expuesto	1	1	1	2	5
557	Tasa de amigabilidad	1	1	1	2	5
558	Juicio de experto sobre amigabilidad	1	1	1	2	5
559	Amigabilidad comparada con la muestra	1	1	1	2	5
560	Tiempo de entrega por lotes	1	1	1	2	5

Núm.	Métrica	Criterios				Total
		Mt1	Mt2	Mt3	Mt4	
561	Capacidad de lotes	1	1	1	2	5
562	Tiempo de procesamiento	1	1	1	2	5
563	Capacidad de procesamiento	1	1	1	2	5
564	Tiempo de transacción interna promedio	1	1	1	2	5
565	Tiempo de transacción interna máximo	1	1	1	2	5
566	Tiempo transcurrido de la CPU	1	1	1	2	5
567	Tiempo de ejecución de la CPU	1	1	1	2	5
568	Tiempo de procesamiento de E/S	1	1	1	2	5
569	Tiempo de procesamiento de la red	1	1	1	2	5
570	Tiempo de procesamiento del terminal	1	1	1	2	5
571	Número de transacciones procesadas	1	1	1	2	5
572	Ocupación de la comunicación	1	1	1	2	5
573	Ocupación de la memoria interna	1	1	1	2	5
574	Ocupación de la memoria externa	1	1	1	2	5
575	Ocupación del procesador	1	1	1	2	5
576	Ocupación de la memoria real	1	1	1	2	5
577	Ocupación de la memoria virtual	1	1	1	2	5
578	Tamaño del WorkingSet	1	1	1	2	5
579	Ocupación del archivo	1	1	1	2	5
580	Ocupación de la red	1	1	1	2	5
581	Utilización de CPU	1	1	1	2	5
582	Utilización de memoria principal	1	1	1	2	5
583	Utilización de canal de E/S	1	1	1	2	5
584	Utilización de archivo	1	1	1	2	5
585	Utilización de red	1	1	1	2	5
586	Utilización de dispositivo de E/S	1	1	1	2	5
587	Utilización de terminal	1	1	1	2	5
588	Proporción de reconocimiento de posición de fallas	1	1	1	2	5
589	Tiempo medio de análisis de falla	1	1	1	2	5
590	Esfuerzo de modificación por unidad de volumen	1	1	1	2	5
591	Esfuerzo de corrección por defecto	1	1	1	2	5
592	Tiempo medio de corrección de falla	1	1	1	2	5
593	Tiempo medio de tratamiento de falla	1	1	1	2	5
594	Tiempo medio de trabajo para corrección de falla	1	1	1	2	5
595	Tiempo de trabajo de revisión significativa por líneas de código cambiadas en el código fuente	1	1	1	2	5
596	Gado de cumplimiento del documento de Mantenimiento	1	1	1	2	5
597	Proporción de nuevas fallas en la revisión	1	1	1	2	5
598	Esfuerzo de prueba por unidad de volumen	1	1	1	2	5
599	Número de casos de prueba por unidad de volumen	1	1	1	2	5
600	promedio de tiempo de trabajo del usuario para	1	1	1	2	5

Núm.	Métrica	Criterios				Total
		Mt1	Mt2	Mt3	Mt4	
	verificar la corrección de error					
601	Tiempo de trabajo medio para la prueba de corrección de fallos	1	1	1	2	5
602	Tiempo de prueba para líneas de código cambiadas	1	1	1	2	5
603	Proporción de control de esfuerzo	1	1	1	2	5
604	Proporción de partes reusables	1	1	1	2	5
605	Proporción de partes rehusadas	1	1	1	2	5
606	Esfuerzo de Portabilidad	1	1	1	2	5
607	Coeficiente aplicable del entorno de hardware	1	1	1	2	5
608	Coeficiente aplicable del entorno de sistema operativo	1	1	1	2	5
609	Coeficiente aplicable de medio ambiente de datos	1	1	1	2	5
610	Coeficiente aplicable de medio ambiente de operación	1	1	1	2	5
611	Proporción de cambio de parámetro	1	1	1	2	5
612	Proporción de recopilación del programa	1	1	1	2	5
613	Proporción de cambio de archivo	1	1	1	2	5
614	Proporción de cambio de lista de salida	1	1	1	2	5
615	Proporción de Conformidad del estándar	1	1	1	2	5
616	Proporción de cambio de función	1	1	1	2	5
617	Proporción de cambio de código fuente	1	1	1	2	5

Tabla 31. Aplicación de criterios a las métricas

Después de la aplicación de los criterios a las métricas la siguiente información en relación al espectro fue obtenida (ver tabla 32):

Espectro	Cantidad
Claro	19
Gris	334
Oscuro	264
Total Métricas	617

Tabla 32. Relación de Cantidad de Métricas en el espectro

ANEXO E: APLICACIÓN DE LOS CRITERIOS A LAS HEURÍSTICAS DE CALIDAD ENCONTRADAS

A continuación esta los criterios de selección aplicados a las heurísticas recopiladas.

- Ht1: *Frecuencia de aparición*: La heurística ha sido tratada por más de un autor.
- Ht2: *Es simple*: La heurística de diseño está escrita en lenguaje simple, claro y entendible de manera que es fácil de usar.
- Ht3: *Exacto*: La heurística está escrita de manera que sea precisa y dirigida hacia un aspecto específico del diseño de software.
- Ht4: *Contradicciones*: La aplicación de la heurística de diseño no contradice la aplicación de otra, salvo si pertenecen a un mismo grupo de heurísticas
- Ht5: *Interpretaciones*: La heurística, en su definición, presenta una y solo una interpretación como recomendación para el diseño de software (es decir, cuál es la recomendación y dónde debe (ría) ser factible aplicarla).

Cada uno de los anteriores es aplicado para una heurística dada, de la siguiente manera:

1: La heurística no satisface el criterio planteado.

2: La heurística satisface el criterio planteado.

Al aplicarlo a cada heurística, es obtenido un puntaje total, siendo 10 el máximo posible. Para que una heurística en particular pueda hacer parte del conjunto final de heurísticas, determinando el puntaje total obtenido, y si éste es igual o superior a 8, la heurística puede ingresar al mencionado conjunto, como es mostrado a continuación (ver tabla 33):

Núm.	Nombre	Criterios					Total
		Ht1	Ht2	Ht3	Ht4	Ht5	
1	Heurística basada en NOS (Número de pasos)	1	2	2	2	2	9
2	Heurística basada en NOAS/NOS (proporción de pasos por actor)	1	2	2	2	2	9
3	Heurística basada en NOSS/NOS (proporción de pasos de sistema)	1	1	2	2	2	8
4	Heurística basada en NOUS/NOS (proporción de pasos de casos de uso)	1	1	2	2	1	7
5	Heurística basada en CC (Complejidad ciclomatica)	1	1	1	2	1	6
6	Chequear el número de pasos	1	2	2	2	2	9
7	Chequear la tasa de diferentes tipos de pasos	1	1	2	2	1	7
8	Chequear el número de excepciones	1	1	2	2	1	7
9	Chequear la Complejidad ciclomatica	1	1	1	2	1	6
10	Encontrar objetos del mundo real	2	2	1	2	1	8
11	Abstracciones consistentes con la forma	2	1	1	1	1	6
12	Encapsular los detalles de Implementación	2	1	1	1	1	6
13	Heredar cuando la jerarquía simplifique	2	2	2	2	2	10

Núm.	Nombre	Criterios					Total
		Ht1	Ht2	Ht3	Ht4	Ht5	
	el diseño						
14	Ocultar secretos (ocultamiento de Información)	2	1	1	2	1	7
15	Secretos y el derecho a la privacidad	2	2	2	2	2	10
16	Barreras para el ocultamiento de información	1	1	1	1	1	5
17	El valor del ocultamiento de la información	1	1	1	2	1	6
18	Identifique áreas probables para cambiar	1	2	1	2	1	7
19	Anticipar diferentes grados de cambio	1	1	1	1	1	5
20	Conserve bajo acoplamiento	2	1	1	2	1	7
21	Apunte hacia fuerte cohesión	2	2	1	1	1	7
22	Construir jerarquías	2	1	1	2	1	7
23	Formalizar contratos de clase	1	1	2	2	2	8
24	Asignar Responsabilidades	1	1	1	1	1	5
25	Diseño por pruebas	2	2	1	2	2	9
26	Elimine fallas	1	1	1	1	1	5
27	Escoja el tiempo de enlace concienzudamente	1	1	1	1	1	5
28	Elaborar puntos centrales de control	1	2	1	1	1	6
29	Considere el uso de la fuerza bruta	1	2	1	1	1	6
30	Dibuje un diagrama	1	2	1	2	2	8
31	Conserve su diseño modular	1	2	2	2	2	9
32	Heurística 2.1	1	2	2	2	2	9
33	Heurística 2.2	1	1	2	2	2	9
34	Heurística 2.3	1	2	2	1	1	7
35	Heurística 2.4	2	2	2	1	2	9
36	Heurística 2.5	2	2	2	2	2	10
37	Heurística 2.6	1	1	1	1	1	5
38	Heurística 2.7	2	1	2	2	2	9
39	Heurística 2.8	2	2	2	2	2	10
40	Heurística 2.9	2	2	2	1	2	9
41	Heurística 2.10	2	2	1	1	1	7
42	Heurística 2.11	1	2	2	2	2	9
43	Heurística 3.1	1	2	1	1	1	6
44	Heurística 3.2	2	2	2	2	2	10
45	Heurística 3.3	2	2	2	2	2	10
46	Heurística 3.4	2	2	2	2	2	10
47	Heurística 3.5	1	2	2	1	1	7

Núm.	Nombre	Criterios					Total
		Ht1	Ht2	Ht3	Ht4	Ht5	
48	Heurística 3.6	2	1	1	1	1	6
49	Heurística 3.7	2	2	2	2	2	10
50	Heurística 3.8	2	2	2	2	2	10
51	Heurística 3.9	1	2	1	2	1	7
52	Heurística 3.10	2	2	1	1	1	7
53	Heurística 4.1	1	2	2	1	1	7
54	Heurística 4.2	1	2	2	2	1	8
55	Heurística 4.3	1	2	1	2	1	7
56	Heurística 4.4	1	2	1	2	1	7
57	Heurística 4.5	2	1	2	2	2	9
58	Heurística 4.6	1	2	1	2	1	7
59	Heurística 4.7	1	2	2	2	2	9
60	Heurística 4.8	1	1	2	1	2	7
61	Heurística 4.9	1	1	1	2	2	7
62	Heurística 4.10	1	1	1	2	2	7
63	Heurística 4.11	2	1	1	2	2	8
64	Heurística 4.12	2	1	1	2	2	8
65	Heurística 4.13	1	2	2	2	2	9
66	Heurística 4.14	1	1	1	2	2	7
67	Heurística 5.1	1	2	2	2	2	9
68	Heurística 5.2	2	2	2	2	2	10
69	Heurística 5.3	2	2	2	2	2	10
70	Heurística 5.4	1	1	2	1	2	7
71	Heurística 5.5	2	2	2	2	2	10
72	Heurística 5.6	1	2	2	2	2	9
73	Heurística 5.7	2	1	1	1	2	7
74	Heurística 5.8	2	2	2	2	2	10
75	Heurística 5.9	2	2	1	2	2	9
76	Heurística 5.10	2	2	2	2	2	10
77	Heurística 5.11	2	1	2	2	2	9
78	Heurística 5.12	2	1	2	2	2	9
79	Heurística 5.13	2	1	2	2	2	9
80	Heurística 5.14	1	1	2	2	2	8
81	Heurística 5.15	1	1	2	1	2	7
82	Heurística 5.16	1	2	2	2	1	8
83	Heurística 5.17	1	1	1	2	2	7
84	Heurística 5.18	1	1	1	2	1	6
85	Heurística 5.19	1	2	2	2	1	8

Núm.	Nombre	Criterios					Total
		Ht1	Ht2	Ht3	Ht4	Ht5	
86	Heurística 6.1	2	2	2	2	2	10
87	Heurística 6.2	2	2	2	2	2	10
88	Heurística 6.3	1	2	2	2	2	9
89	Heurística 7.1	1	2	2	1	2	8
90	Heurística 8.1	2	2	2	2	2	10
91	Heurística 9.1	2	2	2	2	2	10
92	Heurística 9.2	1	2	2	2	2	9
93	Heurística1	1	2	2	2	2	9
94	Heurística2	1	2	1	2	1	7
95	Heurística3	2	2	2	2	1	9
96	Heurística4	2	1	2	1	1	7
97	Heurística5	1	2	1	2	1	7
98	Heurística6	1	1	1	1	1	5
99	Operatividad	1	1	1	1	1	5
100	Capacidad de observación	2	2	1	2	1	8
101	Capacidad de control	1	1	1	1	1	5
102	Capacidad de descomposición	1	2	1	2	1	7
103	Simplicidad	1	1	1	1	1	5
104	Estabilidad	1	1	1	1	1	5
105	Comprensibilidad	1	2	2	2	2	9
106	Apropiabilidad	1	1	1	1	1	5
107	Realimentación	1	2	2	2	1	8
108	Gestionando el error	1	1	1	1	1	5
109	Consistencia	1	2	2	2	2	9
110	Ayuda u orientación	1	2	1	1	1	6
111	Minimizar la carga cognitiva	2	2	2	2	2	10
112	Explícito control de usuario	1	2	1	1	1	6
113	Mapeo natural	1	2	2	2	2	9
114	Accesibilidad	1	2	2	1	1	7
115	Heurística de dinamicidad	1	1	1	1	1	5
116	Usando el modelo de proceso	1	2	1	2	1	7
117	Herencia de Hilos	1	2	1	2	1	7
118	Diseño de clase abstracta	1	2	2	2	2	9
119	Corrección	1	2	2	2	2	9
120	Heurística 1	2	2	2	2	2	10
121	Heurística 2	2	2	2	2	2	10
122	Heurística 3	2	2	2	2	2	10
123	Heurística 4	2	2	2	2	2	10

Núm.	Nombre	Criterios					Total
		Ht1	Ht2	Ht3	Ht4	Ht5	
124	Heurística 5	1	2	1	2	1	7
125	Heurística 6	1	2	2	2	1	8
126	Heurística 7	1	2	2	2	2	9
127	Heurística B1	1	2	2	2	2	9
128	Heurística B2	1	2	2	2	2	9
129	Heurística B3	1	1	1	1	1	5
130	Heurística B4	1	2	2	2	2	9
131	Heurística B5	2	2	2	2	2	10
132	Heurística B6	1	2	2	2	2	9
133	Heurística B7	1	1	1	1	1	5
134	Heurística B8	1	2	2	2	2	9
135	Heurística B9	1	2	2	2	2	9
136	Heurística B10	1	1	2	2	1	7
137	Heurística B11	1	2	2	1	1	7
138	Heurística B12	1	1	1	1	1	5
139	Heurística B13	1	2	2	2	2	9
140	Heurística B14	1	2	2	1	2	8
141	Heurística B15	1	1	1	1	1	5
142	Heurística B16	1	2	1	1	1	6
143	Heurística B17	1	2	1	1	1	6
144	Heurística B18	1	1	1	1	1	5
145	Heurística B19	1	1	1	1	1	5
146	Heurística B20	1	2	2	1	1	7
147	Heurística CA1	1	2	2	2	1	8
148	Heurística CA2	2	2	2	2	1	9
149	Heurística CA3	1	2	2	2	2	9
150	Heurística CA4	1	2	2	1	1	7
151	Heurística CA5	1	1	1	1	1	5
152	Heurística CA6	1	2	2	2	2	9
153	Heurística CA7	1	2	2	1	1	7
154	Heurística CA8	1	2	2	2	2	9
155	Heurística CI1	1	2	2	2	2	9
156	Heurística CI2	2	2	2	2	2	10
157	Heurística CI3	2	2	2	2	2	10
158	Heurística CI4	1	2	1	2	1	7
159	Heurística CI5	1	1	1	1	1	5
160	Heurística CI6	2	1	1	2	1	7
161	Heurística CI7	1	1	1	1	1	5

Núm.	Nombre	Criterios					Total
		Ht1	Ht2	Ht3	Ht4	Ht5	
162	Heurística CI8	1	1	1	1	1	5
163	Heurística CI9	1	1	1	1	1	5
164	Heurística CU1	2	2	2	2	2	10
165	Heurística CU2	1	1	1	2	1	6
166	Heurística CU3	1	2	1	2	1	7
167	Heurística CC1	1	1	1	1	1	5
168	Heurística CC2	2	1	1	2	1	7
169	Heurística CC3	1	1	1	1	1	5
170	Heurística CC4	1	1	1	2	1	6
171	Si hay dependencias de clases concretas	1	2	2	2	2	10
172	Si un objeto tiene diferente comportamiento según su estado	2	2	2	2	2	10
173	Si una jerarquía de clases tiene muchos niveles	2	2	2	2	2	10
174	Si algo es utilizado muy poco o no es utilizado	1	2	2	2	2	9
175	Si una superclase conoce alguna de sus subclases	2	2	2	2	2	10
176	Si una clase colabora con muchas	1	2	2	2	2	9
177	Si un cambio en una interfaz impacta en muchos clientes	1	2	2	2	2	9
178	Si entre una interfaz y su implementación no hay una Abstracción	1	2	2	1	1	7
179	Si una superclase es concreta	2	2	2	2	2	10
180	Si un servicio tiene muchos parámetros	1	2	2	2	2	9
181	Si una clase es grande	2	2	2	2	2	10
182	Si elementos de interfaz de usuario están en entidades de dominio	1	2	2	2	2	9
183	Si una clase utiliza más cosas de otra que de sí misma	1	2	2	2	2	9
184	Si una clase rechaza algo de lo que hereda	1	2	2	2	2	9
185	Si los atributos de una clase son públicos o protegidos	2	2	2	2	2	10

Tabla 33. Aplicación de criterios de calificación por heurística

Debido a que las heurísticas son más bien transversales a los atributos es más complicado clasificarlas en el espectro.

ANEXO F: RELACIÓN ATRIBUTOS-HEURÍSTICAS DE CALIDAD SOFTWARE

A continuación están las posibles relaciones encontradas entre las heurísticas pertenecientes al conjunto final y los atributos de calidad de software. Muestra las relaciones de influencia positiva, dado que las heurísticas buscan favorecer algunos atributos de calidad de software, aunque la aplicación de algunos de ellos no favorezca a otros. En este sentido, aquí sólo son mostradas las relaciones de influencia positiva, porque en el caso contrario, estaría tergiversando el concepto mismo de heurística de calidad de software, que trata de dar una orientación en la resolución de un problema (ver tabla 34).

Núm.	Nombre Heurística	Nombre Atributo
1	Heurística basada en NOS (Número de pasos)	Comprensibilidad Documentación Testeabilidad
2	Heurística basada en NOAS/NOS (proporción de pasos por actor)	Comprensibilidad Documentación Testeabilidad
3	Heurística basada en NOSS/NOS (proporción de pasos de sistema)	Eficiencia Mantenibilidad Reusabilidad Testeabilidad
6	Chequear el número de pasos	Eficiencia Mantenibilidad Reusabilidad Testeabilidad
10	Encontrar objetos del mundo real	Mantenibilidad Reusabilidad Testeabilidad
13	Heredar cuando la jerarquía simplifique el diseño	
15	Secretos y el derecho a la privacidad	Mantenibilidad Reusabilidad Testeabilidad
23	Formalizar contratos de clase	Corrección Mantenibilidad Reusabilidad Testeabilidad
25	Diseño por pruebas	Eficiencia Mantenibilidad Reusabilidad Testeabilidad
30	Dibuje un diagrama	Testeabilidad
31	Conserve su diseño modular	Portabilidad Corrección Eficiencia Mantenibilidad Reusabilidad Testeabilidad
32	Heurística 2.1	Mantenibilidad

Núm.	Nombre Heurística	Nombre Atributo
		Reusabilidad Testeabilidad
33	Heurística 2.2	Mantenibilidad Reusabilidad Testeabilidad
35	Heurística 2.4	Flexibilidad Eficiencia Mantenibilidad Reusabilidad Testeabilidad
36	Heurística 2.5	Mantenibilidad Reusabilidad Testeabilidad
38	Heurística 2.7	Eficiencia Mantenibilidad Reusabilidad Testeabilidad
39	Heurística 2.8	Corrección Mantenibilidad Reusabilidad Testeabilidad
40	Heurística 2.9	Corrección Mantenibilidad Reusabilidad Testeabilidad
42	Heurística 2.11	Corrección Mantenibilidad Reusabilidad Testeabilidad
44	Heurística 3.2	Mantenibilidad Reusabilidad Testeabilidad
45	Heurística 3.3	Corrección Interoperabilidad Mantenibilidad Reusabilidad Testeabilidad
46	Heurística 3.4	Corrección Interoperabilidad Mantenibilidad Reusabilidad Testeabilidad
49	Heurística 3.7	Corrección Mantenibilidad Reusabilidad Testeabilidad
50	Heurística 3.8	Corrección Mantenibilidad Reusabilidad Testeabilidad
54	Heurística 4.2	Eficiencia Mantenibilidad

Núm.	Nombre Heurística	Nombre Atributo
		Reusabilidad Testeabilidad
57	Heurística 4.5	Interoperabilidad Mantenibilidad Reusabilidad Testeabilidad
59	Heurística 4.7	Mantenibilidad Reusabilidad Testeabilidad
63	Heurística 4.11	Corrección Interoperabilidad
64	Heurística 4.12	Mantenibilidad Reusabilidad Testeabilidad
65	Heurística 4.13	Mantenibilidad Reusabilidad Testeabilidad
67	Heurística 5.1	Mantenibilidad Reusabilidad Testeabilidad
68	Heurística 5.2	Mantenibilidad Reusabilidad Testeabilidad
69	Heurística 5.3	Mantenibilidad Reusabilidad Testeabilidad
71	Heurística 5.5	Mantenibilidad Reusabilidad Testeabilidad
72	Heurística 5.6	Mantenibilidad Reusabilidad Testeabilidad
74	Heurística 5.8	Corrección Mantenibilidad Reusabilidad Testeabilidad
75	Heurística 5.9	Corrección Interoperabilidad Mantenibilidad Reusabilidad Testeabilidad
76	Heurística 5.10	Corrección Interoperabilidad Mantenibilidad Reusabilidad Testeabilidad
77	Heurística 5.11	Corrección Interoperabilidad Mantenibilidad Reusabilidad Testeabilidad
78	Heurística 5.12	Mantenibilidad

Núm.	Nombre Heurística	Nombre Atributo
		Reusabilidad Testeabilidad
79	Heurística 5.13	Mantenibilidad Reusabilidad Testeabilidad
80	Heurística 5.14	Mantenibilidad Reusabilidad Testeabilidad
82	Heurística 5.16	Mantenibilidad Reusabilidad Testeabilidad
85	Heurística 5.19	Interoperabilidad Mantenibilidad Reusabilidad Testeabilidad
86	Heurística 6.1	Mantenibilidad Reusabilidad Testeabilidad
87	Heurística 6.2	Testeabilidad
88	Heurística 6.3	Mantenibilidad Reusabilidad Testeabilidad
89	Heurística 7.1	Mantenibilidad Reusabilidad Testeabilidad
90	Heurística 8.1	Mantenibilidad Reusabilidad Testeabilidad
91	Heurística 9.1	Desempeño Corrección Interoperabilidad Mantenibilidad Reusabilidad Testeabilidad
92	Heurística 9.2	Mantenibilidad Reusabilidad Testeabilidad
93	Heurística1	Mantenibilidad Reusabilidad Testeabilidad
95	Heurística3	Mantenibilidad Reusabilidad Testeabilidad
100	Capacidad de observación	Confiabilidad Desempeño Funcionalidad Usabilidad Corrección Testeabilidad
105	Comprensibilidad	Funcionalidad Usabilidad Corrección

Núm.	Nombre Heurística	Nombre Atributo
		Interoperabilidad Testeabilidad
107	Realimentación	Confiabilidad Usabilidad Testeabilidad
109	Consistencia	Confiabilidad Funcionalidad Usabilidad Testeabilidad
111	Minimizar la carga cognitiva	Confiabilidad Usabilidad Testeabilidad
113	Mapeo natural	Confiabilidad Funcionalidad Usabilidad Testeabilidad
118	Diseño de clase abstracta	Flexibilidad Eficiencia Mantenibilidad Reusabilidad Testeabilidad
119	Corrección	Corrección Interoperabilidad Testeabilidad
120	Heurística 1	Flexibilidad Eficiencia Mantenibilidad Reusabilidad Testeabilidad
121	Heurística 2	Flexibilidad Eficiencia Mantenibilidad Reusabilidad Testeabilidad
122	Heurística 3	Eficiencia Mantenibilidad Reusabilidad Testeabilidad
123	Heurística 4	Interoperabilidad Mantenibilidad Reusabilidad Testeabilidad
125	Heurística 6	Mantenibilidad Reusabilidad Testeabilidad
126	Heurística 7	Mantenibilidad Reusabilidad Testeabilidad
127	Heurística B1	Mantenibilidad Reusabilidad Testeabilidad
128	Heurística B2	Mantenibilidad

Núm.	Nombre Heurística	Nombre Atributo
		Reusabilidad Testeabilidad
130	Heurística B4	Mantenibilidad Reusabilidad Testeabilidad
131	Heurística B5	Corrección Eficiencia Mantenibilidad Reusabilidad Testeabilidad
132	Heurística B6	Mantenibilidad Reusabilidad Testeabilidad
134	Heurística B8	Mantenibilidad Reusabilidad Testeabilidad
135	Heurística B9	Mantenibilidad Reusabilidad Testeabilidad
139	Heurística B13	Mantenibilidad Reusabilidad Testeabilidad
140	Heurística B14	Mantenibilidad Reusabilidad Testeabilidad
147	Heurística CA1	Mantenibilidad Reusabilidad Testeabilidad
148	Heurística CA2	Mantenibilidad Reusabilidad Testeabilidad
149	Heurística CA3	Mantenibilidad Reusabilidad Testeabilidad
152	Heurística CA6	Mantenibilidad Reusabilidad Testeabilidad
154	Heurística CA8	Mantenibilidad Reusabilidad Testeabilidad
155	Heurística CI1	Mantenibilidad Reusabilidad Testeabilidad
156	Heurística CI2	Mantenibilidad Reusabilidad Testeabilidad
157	Heurística CI3	Mantenibilidad Reusabilidad Testeabilidad
164	Heurística CU1	Mantenibilidad Reusabilidad Testeabilidad

Núm.	Nombre Heurística	Nombre Atributo
171	Si hay dependencias de clases concretas	Mantenibilidad Reusabilidad Testeabilidad
172	Si un objeto tiene diferente comportamiento según su estado	Mantenibilidad Reusabilidad Testeabilidad
173	Si una jerarquía de clases tiene muchos niveles	Mantenibilidad Reusabilidad Testeabilidad
174	Si algo es utilizado muy poco o no es utilizado	Eficiencia Mantenibilidad Reusabilidad Testeabilidad
175	Si una superclase conoce alguna de sus subclases	Eficiencia Mantenibilidad Reusabilidad Testeabilidad
176	Si una clase colabora con muchas	Flexibilidad Eficiencia Mantenibilidad Reusabilidad Testeabilidad
177	Si un cambio en una interfaz impacta en muchos clientes	Mantenibilidad Reusabilidad Testeabilidad
179	Si una superclase es concreta	Mantenibilidad Reusabilidad Testeabilidad
180	Si un servicio tiene muchos parámetros	Mantenibilidad Reusabilidad Testeabilidad
181	Si una clase es grande	Mantenibilidad Reusabilidad Testeabilidad
182	Si elementos de interfaz de usuario están en entidades de dominio	Mantenibilidad Reusabilidad Testeabilidad
183	Si una clase utiliza más cosas de otra que de sí misma	Mantenibilidad Reusabilidad Testeabilidad
184	Si una clase rechaza algo de lo que hereda	Mantenibilidad Reusabilidad Testeabilidad
185	Si los atributos de una clase son públicos o protegidos	Mantenibilidad Reusabilidad Testeabilidad

Tabla 34. Relación de Heurísticas-Atributos

ANEXO G: ATRIBUTOS DE CALIDAD ENCONTRADOS

Este anexo presenta los atributos encontrados en el transcurso de este proyecto:

La figura 1 muestra la organización de los atributos en capas.

Los términos usados para los atributos son los siguientes:

TITULO EN ESPAÑOL

Nombre en inglés: nombre en idioma ingles

También conocido como: otros nombres con los que es conocido al atributo

Descripción: descripción del atributo más conocida o relevante

Otras definiciones: descripciones hechas por otros autores

Atributos relacionados: relación con otros atributos ya sean internos o externos

Modelos en los que aparece: nombre de los modelos en los que ha sido mencionado

Otros: otras anotaciones.

Como es posible observar, la clasificación indica que el atributo esta en un nivel más alto según su color (espectro), el *claro* (color verde) son aquellos atributos cuya información está más completa, tienen métricas asociadas y son más sencillos de entender; el *gris* (color naranja) indica un nivel intermedio, es decir alguno de los criterios no son cumplidos a cabalidad, el *oscuro* (color rojizo) indica que el atributo está incompleto, esto dificulta su estudio. (ver figura 1)

En el encabezado de cada atributo muestra el color del espectro para el cual ha sido clasificado en el anexo C. A continuación están los atributos encontrados, clasificados por capas:

CAPA 1: FC-ALTO NIVEL

CONFIABILIDAD

Nombre en inglés: Reliability

También conocido como: Fiabilidad

Descripción:

McCall [18]: Referido a las funciones con la precisión requerida. ¿Lo hace acertadamente (de forma fiable) todo el tiempo?

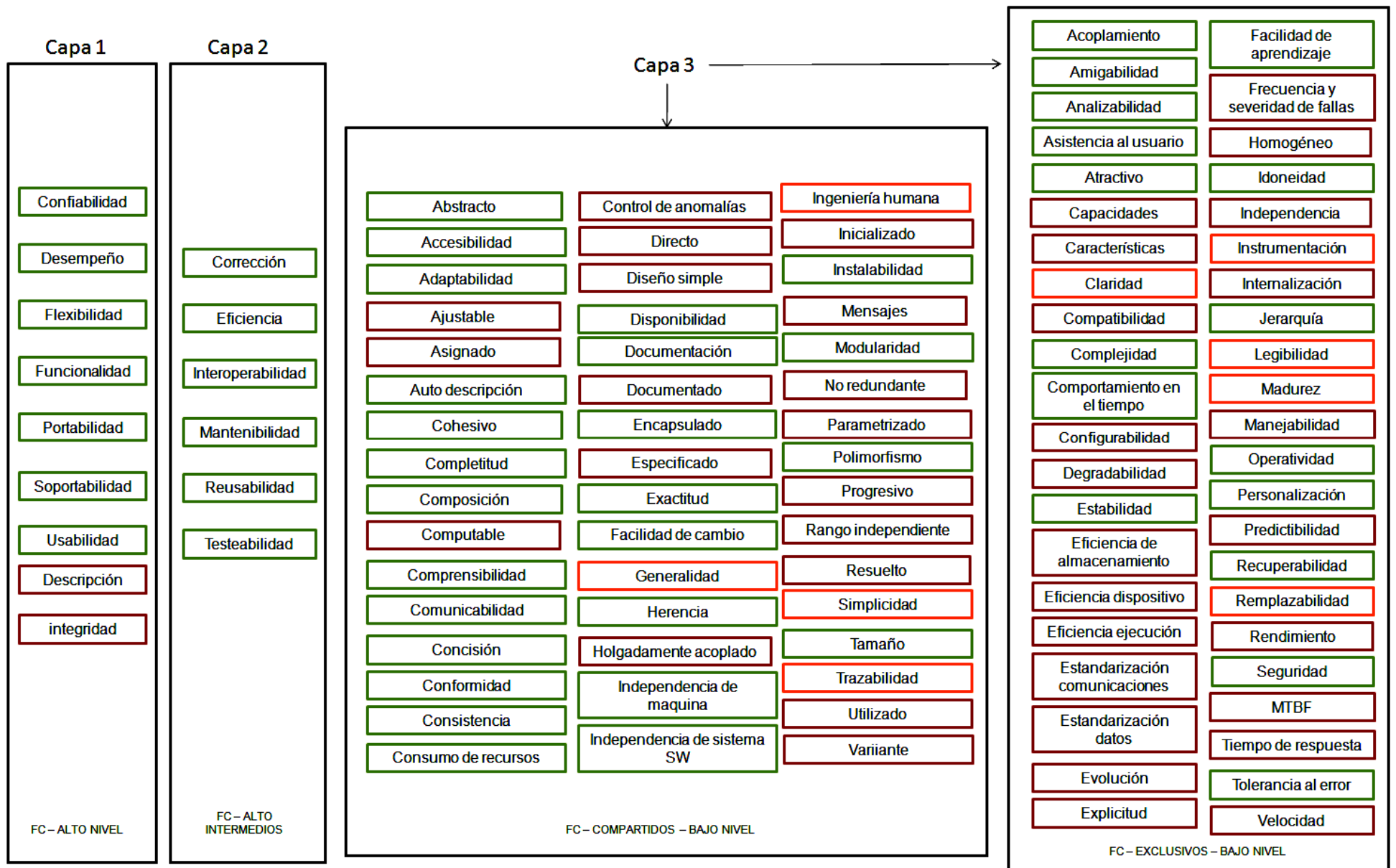


Figura 1. Atributos por capa, basado en “Compilación de un Modelo para Evaluar Atributos de Calidad en Productos Software” [6]

Boehm [17]: Medida en la que el producto software puede ser el esperado para desempeñar sus funciones pretendidas satisfactoriamente.

Otras definiciones:

FURPS [20]: El MTBF¹ del hardware debe ser al menos cinco veces mejor que el sistema de adquisición de datos

ISO 9126 [15]: La capacidad del producto software para mantener su nivel de Desempeño cuando es utilizado bajo condiciones especificadas (durante un determinado período de tiempo).

QUINT2 [58] [59]: Capacidad del software para mantener su nivel de Desempeño bajo condiciones indicadas por un periodo indicado de tiempo

GAFFNEY [63]: Medida de un programa puede ser esperado a realizar su función deseada con la precisión requerida.

ALVARO [64]: La característica expresa la habilidad del componente de mantener un nivel especificado de Desempeño, cuando es usada bajo condiciones específicas.

SOFTWARE QUALITY ASSURANCE GROUP [65]: Medida en que un programa puede esperarse, para llevar a cabo sus funciones deseadas sin fallas por un periodo de tiempo dado.

JETTER [3]: Habilidad de el sistema para no fallar (McCall [18])

PRESSMAN [1].19: Hasta dónde puede esperarse que un programa lleve a cabo su función con la exactitud requerida. Hay que hacer notar que hay propuestas otras definiciones de fiabilidad más completas

ABRIL [34]: Es la capacidad de un producto software de mantener su nivel de Desempeño, bajo condiciones establecidas, por un periodo de tiempo.

OLMEDILLA [53]: La capacidad del software de mantener un nivel específico de rendimiento bajo determinadas condiciones de uso.

IEEE 610.12 [23]: La habilidad de un sistema o componente de realizar sus funciones requeridas bajo condiciones establecidas en un periodo especificado de tiempo.

Atributos relacionados: Consistencia, Tolerancia al error, Exactitud, Simplicidad, Completitud, Madurez, Recuperabilidad, Disponibilidad, Degradabilidad, Conformidad, Frecuencia y severidad de fallas, Tiempo medio entre fallas, Predictibilidad, Encapsulamiento, Composición, Polimorfismo.

Modelos en los que aparece: McCall [18], Boehm [17], FURPS [20], ISO 9126 [15], Dromey [26], QUINT2 [58] [59]

DESEMPEÑO

Nombre en inglés: Performance

También conocido como: Rendimiento

Descripción:

FURPS [20]: La producción total debería ser que al menos deben existir diez veces soluciones para operaciones remotas

Otras definiciones:

¹ MTBF: Middle Time between Failures. Tiempo Medio Entre Fallos. Ver Métrica 331
Página 248.

IEEE 610.12 [23]: El grado en que un sistema o componente logra sus funciones designadas dentro de restricciones dadas, tal como velocidad, exactitud, o uso de memoria.

Atributos relacionados: Velocidad, Tiempo de respuesta, Consumo de recursos, Rendimiento, Eficiencia

Modelos en los que aparece: FURPS [20]

FLEXIBILIDAD

Nombre en inglés: Flexibility

También conocido como:

Descripción:

McCall [18]: Esfuerzo necesario para modificar un programa operativo. ¿Puedo cambiarlo?

Otras definiciones:

Bansiya [31]: Características que permiten la incorporación de cambios en un diseño para ser adaptado para proveer capacidades relacionadas con Funcionalidad

BERANDER [36]: La Facilidad con la cual un sistema o componente puede ser modificado para uso en aplicaciones u otros entornos, para los que fueron diseñados específicamente

GAFFNEY [63], SOFTWARE QUALITY ASSURANCE GROUP [65]: Esfuerzo requerido para modificar un programa operacional.

PRESSMAN [1]: El esfuerzo necesario para modificar un programa que ya está en funcionamiento.

MCCONNELL [66], IEEE 610.12 [23]: La Facilidad con la cual un sistema o componente puede ser modificado para usar en otras aplicaciones o entornos, para los cuales fue específicamente diseñados.

Atributos relacionados: Modularidad, generalidad, Facilidad de cambio, Auto descripción, Encapsulamiento, Composición, Polimorfismo

Modelos en los que aparece: McCall [18], Bansiya [31]

Otros: A pesar de que algunos autores como IEEE 610.12 [23] consideran a flexibilidad igual a adaptabilidad en este trabajo son considerados diferentes.

FUNCIONALIDAD

Nombre en inglés: Functionality

También conocido como: Facilidad de operación

Descripción:

FURPS [20]: Todas las funciones del sistema de adquisición de datos debe operar de la misma forma como lo hacen localmente

Otras definiciones:

ISO 9126 [15]: La capacidad del producto software para proporcionar las funciones que satisfacen las necesidades establecidas e implícitas cuando el software es utilizado bajo condiciones especificadas

Bansiya [31]: Las responsabilidades asignadas para las clases en un diseño, las que son hechas disponibles por las clases a través de sus interfaces públicas

QUINT2 [58] [59]: Existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son aquellas que satisfacen necesidades dichas o implícitas

ALVARO [64]: Esta característica expresa la capacidad de un componente de mantener un nivel especificado de Desempeño, cuando es usado bajo condiciones específicas.

ABRIL [34]: Es la capacidad de un producto software de satisfacer los requisitos funcionales prescriptos y las necesidades implícitas de los usuarios.

OLMEDILLA [53]: La capacidad del software de proveer las funciones que cumplen con las necesidades implícitas y explícitas cuando el mismo es utilizado bajo ciertas condiciones.

Atributos relacionados: Tamaño de diseño, Jerarquía, Cohesión, Polimorfismo, Mensajes, Idoneidad, Exactitud, Interoperabilidad, Seguridad, Trazabilidad, Conformidad, Características, Capacidades, Generalidad.

Modelos en los que aparece: FURPS [20], ISO 9126 [15], Dromey [26], Bansiya [31], QUINT2 [58] [59]

PORTABILIDAD

Nombre en inglés: Portability

También conocido como: TransPortabilidad, Auto contención

Descripción:

McCall [18]: Esfuerzo requerido para transferir un programa desde una configuración de hardware y / o el entorno de un sistema software a otro. ¿Podré usarlo en otra máquina?

Boehm [17]: Un producto de software posee la característica de la Portabilidad en la medida en que puede manejar fácilmente y bien en otras configuraciones de equipo que no sea la actual.

Otras definiciones:

ISO 9126 [15]: La capacidad del producto software para ser transferido de un ambiente a otro

QUINT2 [58] [59]: Habilidad del software para ser transferido de un ambiente a otro.

BERANDER [36]: La Facilidad con la que sistema o componente puede ser transferido desde un entorno hardware o software a otro.

GAFFNEY [63], PRESSMAN [1]: Esfuerzo requerido para transferir un programa desde una configuración hardware y/o configuración de sistema software a otro.

ALVARO [64]: Esta característica es definida como la habilidad de un componente a ser transferencia desde un entorno a otro.

SOFTWARE QUALITY ASSURANCE GROUP [65], OLMEDILLA [53]: Esfuerzo requerido para transportar el software para uso en otro entorno.

ABRIL [34]: Es la capacidad de un producto software de ser transferido de un ambiente a otro. *Nota:* El ambiente puede ser organizacional, de software o de hardware

IEEE 610.12 [23]: La Facilidad con la cual un sistema software o componente puede ser transferido desde un entorno hardware o software a otro. Syn: TransPortabilidad

MCCONNELL [66]: La Facilidad con que puede modificarse un sistema para operar en un entorno diferente del que fue específicamente diseñado.

Atributos relacionados: Modularidad, Auto descripción, Independencia de máquina, Independencia de sistema software, Completitud, Adaptabilidad, Instalabilidad, Coexistencia, Remplazabilidad, Conformidad, Modularidad, Independencia, Documentación

Modelos en los que aparece: McCall [18], Boehm [17], ISO 9126 [15], Dromey [26], SQA [30], QUINT2 [58] [59]

Soportabilidad

Nombre en inglés: Supportability

También conocido como: Compatibilidad

Descripción:

FURPS [20]: Los tableros deben ser capaces de un auto test automático directo, así como también un auto test de computador inicializado

Otras definiciones:

IEEE 610.12 [23]: (1) La habilidad de dos o más sistemas de realizar sus funciones requeridas mientras comparten el mismo entorno hardware o software. (2) La habilidad de dos o más sistemas o componentes de intercambiar información.

Atributos relacionados: Testabilidad, Facilidad de cambio, Adaptabilidad, Mantenibilidad, Compatibilidad, Configurabilidad, Asistencia de usuario, Instalabilidad, Internalización.

Modelos en los que aparece: FURPS [20]

USABILIDAD

Nombre en inglés: Usability

También conocido como: Facilidad de uso, As utility (Boehm [17])

Descripción:

McCall [18]: Esfuerzo necesario para aprender, operar, preparar la entrada, e interpretar la salida de un programa. ¿Puedo usarlo?

Boehm [17]: Medida en la que producto es conveniente y practicable para usar. La interface humana es importante aun si el producto no tiene utilidad general

Otras definiciones:

FURPS [20]: No debería ser necesario cambios para programas de aplicación de usuarios

ISO 9126 [15]: La capacidad del producto software para ser comprendido, aprendido, utilizado y que sea atractivo para el usuario, cuando es utilizado bajo condiciones especificadas.

QUINT2 [58] [59]: Esfuerzo necesitado por uso, y estimación individual de tal uso, por un conjunto de usuarios indicado o implícito

GAFFNEY [63], SOFTWARE QUALITY ASSURANCE GROUP [65], PRESSMAN [1]: Esfuerzo requerido para aprender a operar, preparar entradas, e interpretar salidas del programa.

ALVARO [64], OLMEDILLA [53]: Esta característica expresa la habilidad de un componente para ser entendido, aprendido, usado, configurado, y ejecutado, cuando es usado bajo condiciones especificas.

IEEE 610.12 [23]: La Facilidad con la cual un usuario puede aprender para operar, preparar entradas para e interpretar salidas de un sistema o componente.

MCCONNELL [66]: La Facilidad con la cual los usuarios pueden aprender y usar un programa.

Atributos relacionados: Entrenamiento, Comunicatividad, operatividad, Factores humanos, estética, consistencia, documentación, Confiabilidad, Eficiencia, ingeniería humana, Comprensibilidad, Facilidad de aprendizaje, atractivo, Conformidad, amigabilidad, explicitud, personalización, claridad, asistencia de usuario

Modelos en los que aparece: McCall [18], Boehm [17], FURPS [20], ISO 9126 [15], Dromey [26], QUINT2 [58] [59]

DESCRIPCIÓN

Nombre en inglés: Descriptiveness

Atributos relacionados: auto descripción, documentación

Modelos en los que aparece: SQA [30]

INTEGRIDAD

Nombre en inglés: Integrity

Descripción:

McCall [18]: Grado en que el acceso a programas o datos por personas no autorizadas pueden ser controlados. ¿Es seguro?

Otras definiciones:

JETTER [3]: La protección del programa para acceso no autorizado.

PRESSMAN [1]: Hasta dónde puede controlarse el acceso al software o a los datos por personas no autorizadas

IEEE 610.12 [23]: el grado en el que un sistema o componente previene acceso no autorizado, o modificación de programas del computador o datos.

MCCONNELL [66]: el grado en que un sistema previene acceso no autorizado o impropio a sus programas y datos. La idea de integridad incluye restricción de acceso a usuarios no autorizados así como también que los datos son accedidos apropiadamente.

MEYER [62]: es la habilidad del sistema software de proteger sus diversos componentes (Programas y datos) contra acceso no autorizado y modificación

Atributos relacionados: Control de acceso, auditoria de acceso

Modelos en los que aparece: McCall [18]

CAPA 2: FC-INTERMEDIOS

CORRECCIÓN

Nombre en inglés: Correctness

Descripción:

McCall [18]: Medida en que un programa satisface sus especificaciones y cumple con los objetivos de la misión del usuario. La función que el software está realizando es incorrecta, la evaluación esta en términos del esfuerzo requerido para arreglarlo. ¿Hace lo que quiero?

Otras definiciones:

Dromey [26]: propiedades importantes, cuya violación afecta el rendimiento del producto.

BERANDER [36]: pg. 54: el grado en el cual un sistema o componente es libre de fallas en su especificación, diseño e implementación.

GAFFNEY [63]: medida en que un programa satisface sus especificaciones y cumple a cabalidad los objetivos de la misión del usuario.

ALVARO [64]: este atributo evalúa el porcentaje de los resultados obtenidos con precisión, especificados por los requerimientos de usuario.

JETTER [3]: medida en que un programa conforma su especificación.

IEEE 610.12 [23]: (1) el grado en el que un sistema o componente es libre de fallas en su especificación, diseño e implementación. (2) el grado en que el software, documentación, u otros ítems encuentran requerimientos especificados. (3) el grado en que el software, documentación, u otros ítems encuentran necesidades de usuario.

SOFTWARE QUALITY ASSURANCE GROUP [65]: medida en que un software satisface sus especificaciones y cumple a cabalidad los objetivos del usuario.

MEYER [62]: la habilidad de los productos software para realizar sus tareas exactas, como fueron definidas en su especificación.

PRESSMAN [1]-19: hasta donde satisface un programa su especificación y logra los objetivos propuestos por el cliente.

Atributos relacionados: Trazabilidad, Consistencia, Completitud, Funcionalidad, Confiabilidad

Modelos en los que aparece: McCall [18], Dromey [26]

EFICIENCIA

Nombre en inglés: Efficiency

También conocido como: Efectividad (Bansiya [31]), eficacia, efectivo (Dromey [26])

Descripción:

McCall [18]: La cantidad de recursos informáticos y de código requeridos por un programa para realizar una función. ¿Funcionara en mi hardware tan bien como pueda?

Boehm [17]: Medida en la que un producto software cumple a cabalidad su propósito sin malgastar recursos

Otras definiciones:

ISO 9126 [15]: La capacidad del producto software para proporcionar un Desempeño apropiado, en relación con la cantidad de recurso utilizado, bajo condiciones establecidas

Bansiya [31]: Este refiere a la habilidad del diseño para conseguir la Funcionalidad y comportamiento deseados usando conceptos de diseño Orientado a Objetos y técnicas.

QUINT2 [58] [59]: Relación entre el nivel de Desempeño del software y la cantidad de recursos usados, bajo condiciones indicadas

GAFFNEY [63], SOFTWARE QUALITY ASSURANCE GROUP [65], PRESSMAN [1]-19: La cantidad de recursos computacionales y de código requeridos por el software para realizar una función.

ALVARO [64]: La característica expresa la habilidad de un componente para proveer un Desempeño apropiado, relativo a la cantidad de recursos usados.

ABRIL [34]: Es la capacidad de un producto software de proporcionar un rendimiento apropiado, de acuerdo a la cantidad de recursos usados bajo condiciones establecidas.

OLMEDILLA [53]: la capacidad del software de ofrecer el rendimiento apropiado con respecto a la cantidad de recursos utilizados, bajo condiciones prefijadas.

IEEE 610.12 [23]: el grado en el que un sistema o componente realiza sus funciones designadas, con un mínimo consumo de recursos.

MCCONNELL [66]: Mínimo uso de los recursos del sistema, incluyendo memoria y tiempo de ejecución.

Atributos relacionados: Eficiencia de ejecución, Eficiencia de almacenamiento, Eficiencia de dispositivo, accesibilidad, Desempeño, comportamiento en el tiempo, utilización de recursos, Conformidad, Abstracción, Encapsulamiento, composición, Herencia, Polimorfismo

Modelos en los que aparece: McCall [18], Boehm [17], FURPS [20], ISO 9126 [15], Dromey [26], Bansiya [31], QUINT2 [58] [59]

INTEROPERABILIDAD

Nombre en inglés: Interoperability

Descripción:

McCall [18]: Esfuerzo necesario para acoplar un sistema con otro. ¿Podré hacerlo interactuar con otro sistema?

ISO 9126 [15]: La capacidad del producto de software para interactuar con uno o más sistemas específicos. (Usa la Interoperabilidad en lugar de compatibilidad para evitar posibles ambigüedades con reemplazabilidad).

Otras definiciones:

QUINT2 [58] [59]: Habilidad para interactuar con sistemas específicos

BERANDER [36], IEEE 610.12 [23]: la habilidad de dos o más sistemas o componentes para intercambiar información y usar la información que ha sido intercambiada.

GAFFNEY [63], JETTER [3], PRESSMAN [1]: esfuerzo requerido para acoplar un sistema con otro.

SOFTWARE QUALITY ASSURANCE GROUP [65]: medida en que un software puede ser usado en otras aplicaciones.

Atributos relacionados: Modularidad, Estandarización con las comunicaciones, Estandarización con los datos, Funcionalidad

Modelos en los que aparece: McCall [18], ISO 9126 [15], QUINT2 [58] [59]

MANTENIBILIDAD

Nombre en inglés: Maintainability

También conocido como: Facilidad de Mantenimiento

Descripción:

McCall [18]: esfuerzo necesario para localizar y corregir un error en un programa operativo. ¿Puedo arreglarlo?

Boehm [17]: en la que un producto facilita la actualización para satisfacer nuevos requerimientos

Otras definiciones:

ISO 9126 [15]: La capacidad del producto software para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptaciones del software a cambios de ambiente y en requisitos y especificaciones funcionales

QUINT2 [58] [59]: Esfuerzo necesitado para hacer modificaciones específicas.

GAFFNEY [63], SOFTWARE QUALITY ASSURANCE GROUP [65], JETTER [3], PRESSMAN [1]: esfuerzo requerido para localizar y arreglar un error un programa operacional

ALVARO [64]: esta característica describe la habilidad de un componente para ser modificado.

ABRIL [34]: Es la capacidad de un producto software para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación del software a cambios en el entorno, en los requisitos o en las especificaciones funcionales.

OLMEDILLA [53]: la capacidad del producto de ser modificado. Dichas modificaciones pueden incluir correcciones, mejoras o adaptaciones a cambios en el entorno y en los requisitos y especificaciones funcionales.

IEEE 610.12 [23]: (1) la Facilidad con la cual un sistema software o componente puede ser modificado para corregir fallas, o aptado para un entorno cambiado. (2) la Facilidad con la cual un sistema hardware o componente puede retener, o reparar un estado en el cual pueda realizar sus funciones requeridas.

MCCONNELL [66]: la Facilidad con la cual usted puede modificar un sistema software para cambiar o adicionar capacidades, mejorar rendimiento, o corregir defectos.

Atributos relacionados: consistencia, simplicidad, concisión, modularidad, auto descripción, Testeabilidad, Comprensibilidad, Modificabilidad, Soportabilidad, Analizabilidad, Facilidad de cambio, Estabilidad, Testeabilidad, Conformidad, diseño simple, control de anomalías, documentación, manejabilidad, Reusabilidad

Modelos en los que aparece: McCall [18], Boehm [17], FURPS [20], ISO 9126 [15], Dromey [26], SQA [30], QUINT2 [58] [59]

REUSABILIDAD

Nombre en inglés: Reusability

También conocido como: Facilidad de Reutilización

Descripción:

McCall [18]: Medida en que un programa puede ser utilizado en otras aplicaciones – relacionado a empaquetamiento y el alcance de las funciones que realizan los programas. ¿Seré capaz de rehusar algo del software?

Boehm [17]: Refleja la presencia de características de diseño Orientado a objetos que permite un diseño para ser re aplicado a un nuevo problema sin esfuerzo significativo

Otras definiciones:

QUINT2 [58] [59]: Potencial para rehusar completo o parcialmente en otros productos software

BERANDER [36], IEEE 610.12 [23]: el grado en el que un modulo software u otro producto de trabajo puede ser usado en más de un programa de computo o sistema software.

GAFFNEY [63]: medida en que un programa puede ser usado en otras aplicaciones reaccionadas a empaquetamiento y alcance de las funciones que los programas realizan.

SOFTWARE QUALITY ASSURANCE GROUP [65]: medida en que un software puede ser usado en otras aplicaciones.

MEYER [62], MCCONNELL [66]: habilidad de elementos software para servir en la construcción de muchas diferentes aplicaciones.

PRESSMAN [1]: (capacidad de Reutilización). Hasta dónde puede volverse a emplear un programa (o partes de un programa) en otras aplicaciones, en relación al empaquetamiento y alcance de las funciones que realiza el programa

Atributos relacionados: Generalidad, modularidad, independencia de sistema software, independencia de máquina, auto descripción, Tamaño de diseño, Cohesión, Mensajes, Mantenibilidad

Modelos en los que aparece: McCall [18], Dromey [26], Bansiya [31], QUINT2 [58] [59]

TESTEABILIDAD

Nombre en inglés: Testability

También conocido como: Facilidad de prueba

Descripción:

McCall [18]: Esfuerzo requerido para probar un programa para asegurar que cumple su función. ¿Puedo probarlo?

Boehm [17]: Medida en la que un producto facilita el establecimiento de criterios de aceptación y evaluación de soporte para su Desempeño

Otras definiciones:

ISO 1926: La capacidad del producto software para permitir validar las modificaciones

QUINT2 [58] [59]: Esfuerzo necesitado para autenticar el software (modificado)

BERANDER [36]: el grado en el cual un sistema o componente facilita el establecimiento de criterios de prueba y la realización de pruebas para determinar si esos criterios han sido encontrados.

GAFFNEY [63], PRESSMAN [1]: esfuerzo requerido para probar un programa asegurar la realización de su función deseada.

IEEE 610.12 [23]: (1) El grado en el cual un sistema o componente facilita el establecimiento de criterios de prueba y la realización de prueba para determinar si esos criterios han sido encontrados. (2) el grado de para el cual un requerimiento está indicado en términos que permitan el establecimiento de criterios de prueba y realizar pruebas para determinar que esos criterios han sido encontrados.

Atributos relacionados: Simplicidad, modularidad, instrumentación, auto descripción, accesibilidad, Comunicatividad, Estructurado, Soportabilidad, Mantenibilidad

Modelos en los que aparece: McCall [18], Boehm [17], FURPS [20], ISO 9126 [15], QUINT2 [58] [59]

CAPA3: FC-COMPARTIDOS-BAJO NIVEL

ABSTRACTO

Nombre en inglés: Abstract

También conocido como: Abstracción (Bansiya [31])

Descripción:

Dromey [26]: Un objeto/modulo es suficientemente abstracto si no hay un obvio, útil concepto de alto nivel que abarque la forma estructural.

Bansiya [31]: Una medida aspectos de Generalización especialización en el diseño. Las clases en el diseño que tienen uno o más descendientes exhibidos en esta propiedad de Abstracción

Otras definiciones:

IEEE 610.12 [23]: Abstracción: una vista de un objeto que es enfocada sobre la información relevante de un propósito particular e ignora el resto de la información.

Atributos relacionados: Mantenibilidad, Reusabilidad, Facilidad de cambio

Modelos en los que aparece: Dromey [26], Bansiya [31]

Otros: La Abstracción puede definirse como "la capacidad para examinar algo sin preocuparse por sus datos internos". Por su parte, abstracto refiere a una característica del software, e.g., clase abstracta.

ACCESIBILIDAD

Nombre en inglés: Accessibility

También conocido como: Facilidad de acceso, Control de acceso (McCall [18])

Descripción:

Boehm [17]: Medida en la que un producto software facilita el uso selectivo de sus componentes

McCall [18]: atributos de software que son proporcionados para control de acceso de software y datos

Otras definiciones:

SOFTWARE QUALITY ASSURANCE GROUP [65]: las características del software que proporcionan para control y auditoria de acceso al software y los datos.

Atributos relacionados: Ingeniería humana, Eficiencia, Testeabilidad

Modelos en los que aparece: Boehm [17], McCall [18]

ADAPTABILIDAD

Nombre en inglés: Adaptability

Descripción:

ISO 9126 [15]: La capacidad del producto software para adaptarse a diferentes ambientes especificados sin aplicar acciones o medios distintos a los ofrecidos para este propósito por el software considerado

Otras definiciones:

QUINT2 [58] [59]: Oportunidad para la adaptación del software a diferentes ambientes especificados sin la aplicación de otras acciones o métodos que ellos proveen para este propósito del software en cuestión

MCCONNELL [66]: medida en que los sistemas pueden ser usados, sin modificación, en aplicaciones o entornos aparte de para los cuales fueron específicamente diseñados.

Atributos relacionados: Portabilidad, Soportabilidad

Modelos en los que aparece: FURPS [20], ISO 9126 [15], QUINT2 [58] [59]

AUTO DESCRIPCIÓN

Nombre en inglés: Self-descriptiveness

También conocido como: Especificado (Dromey [26])

Descripción:

McCall [18]: atributos de software que proporcionan explicación de la implementación de una función

Boehm [17]: Medida en la que un producto software contiene información suficiente para un lector para determinar los objetivos, suposiciones, restricciones, entradas, salidas, componentes y estado del producto

Otras definiciones:

SQAE [30]: ¿La documentación incrustada, convenciones de nombre, etc., proveen suficiente y breve comprensión dentro del funcionamiento del propio código?

Dromey [26]: Una forma estructural es auto descriptiva si su propósito, estrategia, determinación, o propiedades son claramente evidentes desde la elección de nombres para módulos y varios identificadores son significativos y congruentes con el contexto de la aplicación.

SOFTWARE QUALITY ASSURANCE GROUP [65]: Las características del software que proporcionan explicación de la implementación de una función.

PRESSMAN [1]: el grado en que el código fuente proporciona documentación significativa.

Atributos relacionados: Mantenibilidad, flexibilidad, Testeabilidad, Portabilidad, Reusabilidad, Comprensibilidad, Evolución, descripción, Usabilidad

Modelos en los que aparece: McCall [18], Boehm [17], Dromey [26], SQAE [30]

Otros: esta relacionado con el concepto de auto documentación

COHESIÓN

Nombre en inglés: Cohesion

También conocido como: Cohesivo

Descripción:

Dromey [26]: Una forma estructural es cohesiva si todos sus elementos son delimitados apretadamente uno al otro y todos ellos contribuyen a conseguir un objetivo simple o función.

Bansiya [31]: Evaluar la relevancia de métodos y atributos en una clase. Fuerte traslapado en los parámetros de método y tipo de atributo es una indicación de cohesión fuerte

Otras definiciones:

IEEE 610.12 [23]: la forma y grado en que las tareas realizadas por un simple modulo de software es relacionado con otro. Tipos incluyen coincidencial, comunicacional, funcional, lógico, procesal, secuencial, y temporal.

Atributos relacionados: Mantenibilidad, Reusabilidad, Comprensibilidad, Funcionalidad

Modelos en los que aparece: Dromey [26], Bansiya [31]

COMPLETITUD

Nombre en inglés: Completeness

También conocido como: compleción.

Descripción:

McCall [18]: los atributos del software que proporcionan una total implementación de las funciones requeridas

Boehm [17]: Medida en la que todas las partes un producto están presentes, y están plenamente desarrolladas.

Otras definiciones:

Dromey [26]: Cuando una forma estructural tiene todos los elementos necesarios para definir e implementar la forma estructural así que esta pueda cumplir a cabalidad sus roles pretendido en una manera que no impacte la fiabilidad o Funcionalidad

ALVARO [64]: es posible que algunas implementaciones no cubran completamente los servicios especificados. Este atributo mide el número de operaciones implementadas comparado al número total de operaciones especificadas.

SOFTWARE QUALITY ASSURANCE GROUP [65]: la característica del software que proporciona una implementación completa de las funciones requeridas.

PRESSMAN [1] El grado con que es lograda la implementación total de una función

Atributos relacionados: Corrección, Portabilidad, Confiabilidad, Funcionalidad, Mantenibilidad, Usabilidad

Modelos en los que aparece: McCall [18], Boehm [17], Dromey [26]

COMPOSICIÓN

Nombre en inglés: Composition

Descripción:

Bansiya [31]: Medidas de relaciones "Parte-de", "tener", "Consiste-en" o "parte entera", las cuales son relaciones de agregación en diseño Orientado a Objetos

Atributos relacionados: Confiabilidad, Eficiencia

Modelos en los que aparece: Bansiya [31]

COMUNICABILIDAD

Nombre en inglés: Communicativeness

También conocido como: Comunicatividad

Descripción:

McCall [18]: Atributos de software que proporcionan entradas útiles y salidas que pueden ser asimiladas.

Otras definiciones:

Boehm [17]: Medida en la que un producto software facilita la especificación de entradas y provee salidas cuyas formas y contenido son fáciles de asimilar y usables

Atributos relacionados: Usabilidad, Ingeniería humana, Testeabilidad

Modelos en los que aparece: McCall [18], Boehm [17]

CONCISIÓN

Nombre en inglés: Conciseness

Descripción:

McCall [18]: Atributos de software que proporcionan para la implementación de una función con una cuenta mínima de código.

Otras definiciones:

Boehm [17]: En la que no hay exceso de información

PRESSMAN [1]: Lo compacto que es el programa en términos de líneas de código.

Atributos relacionados: Mantenibilidad, Comprensibilidad

Modelos en los que aparece: McCall [18], Boehm [17]

CONFORMIDAD

Nombre en inglés: Conformance

También conocido como: Facilidad de auditoría (McCall [18]), Auditora de acceso (Access audit - McCall [18])

Descripción:

ISO 9126 [15]: La capacidad del producto software para adherirse a estándares, convenciones o regulaciones relacionadas con los atributos.

Otras definiciones:

QUINT2 [58] [59]: Adhesión a estándares o convenciones

McCall [18]: Atributos de software que son proporcionados para una auditoría del acceso de software y datos

PRESSMAN [1]: La Facilidad con la que puede comprobarse el cumplimiento de los estándares.

IEEE 610.12 [23]: Facilidad de auditoría: una exámen independiente de un producto de trabajo o conjunto de productos de trabajo para asegurar Conformidad con especificaciones, estándares, contrato contractual, u otros criterios

Atributos relacionados: Portabilidad, Usabilidad, Confiabilidad, Confiabilidad, Eficiencia, Funcionalidad, Mantenibilidad, Interoperabilidad

Modelos en los que aparece: ISO 9126 [15], QUINT2 [58] [59]

Otros: En este atributo es considerado que incluye a Facilidad de auditoría, Estandarización en las comunicaciones, Estandarización en los datos, Conformidad con la Confiabilidad, Conformidad con la Eficiencia, Conformidad con la Funcionalidad, Conformidad con la Mantenibilidad, Conformidad con la Portabilidad

CONSISTENCIA

Nombre en inglés: Consistency

También conocido como: CoHerencia, Consistente (Dromey [26])

Descripción:

McCall [18]: Los atributos del software que proporcionan un diseño uniforme, técnicas de implementación y notación

Boehm [17]: Medida en la que existe uniformidad en notación, terminología, y dentro de la simbología propia. Externa: Medida en la que el contenido es trazable hacia los requerimientos

Otras definiciones:

Dromey [26]: Una forma estructura es consistente si su uso mantiene sus propiedades o Funcionalidad y todos sus elementos contribuyen, a y refuerzan sus objetivos conjuntos o efectos

SQAE [30]: ¿Tienen los productos del proyecto (código y documentación) están contruidos con un estilo uniforme para un estándar documentado?

SOFTWARE QUALITY ASSURANCE GROUP [65]: Las características del software que proporcionan técnicas y notación uniformes de diseño e implementación

PRESSMAN [1]: El empleo de un diseño uniforme y de técnicas de documentación a lo largo del proyecto de desarrollo del software.

IEEE 610.12 [23]: El grado de uniformidad, estandarización y libertad de contradicción entre los documentos o partes de un sistema o componente.

Atributos relacionados: Corrección, Confiabilidad, Mantenibilidad, Funcionalidad, Usabilidad, Portabilidad, Re-usabilidad

Modelos en los que aparece: McCall [18], Boehm [17], FURPS [20], Dromey [26], SQAE [30]

CONSUMO DE RECURSOS

Nombre en inglés: Resource usage, Resource utilization

También conocido como: Utilización de recursos (ISO 9126 [15], QUINT2 [58] [59])

Descripción:

ISO 9126 [15]: La capacidad del producto software para utilizar una apropiada cantidad y tipos de recursos cuando el software desempeña su función bajo condiciones establecidas

Otras definiciones:

QUINT2 [58] [59]: Cantidad de recursos usados y la duración de cada uso desempeñando funciones del software

Atributos relacionados: Eficiencia, Desempeño

Modelos en los que aparece: FURPS [20], ISO 9126 [15], QUINT2 [58] [59]

DISPONIBILIDAD

Nombre en inglés: Availability

Descripción:

QUINT2 [58] [59]: Cantidad de tiempo de disponibilidad del producto para el usuario en el tiempo que este lo necesita.

Otras definiciones:

IEEE 610.12 [23]: El grado en que un sistema o componente es operacional y accesible cuando es requerido para uso. A menudo expresado como probabilidad.

Atributos relacionados: Confiabilidad

Modelos en los que aparece: QUINT2 [58] [59]

DOCUMENTACIÓN

Nombre en inglés: Documentation

También conocido como: Documentado (Dromey [26])

Descripción:

IEEE 610.12 [23]: (1) Una colección de documentos de un determinado tema. (2) Cualquier información escrita o gráfica describiendo, definiendo, especificando,

reportando, o certificando actividades, requerimientos, procedimientos o resultados. (3) El proceso de generar o revisar un documento. (4) La administración de documentos, incluyendo la identificación, adquisición, procesamiento almacenamiento y difusión.

Otras definiciones:

Dromey [26]: Una forma estructural es documentada si su propósito, estrategia, determinación y propiedades son todas explícita y precisamente definidas dentro del contexto de la forma estructural.

SQAE [30]: ¿La copia de documentación es adecuada para el soporte de Mantenimiento, puerto, incremento y reingeniería par el proyecto?

Atributos relacionados: Usabilidad, Mantenibilidad, Evolución, Portabilidad, Descripción, Portabilidad, Re-usabilidad

Modelos en los que aparece: FURPS [20], Dromey [26], SQAE [30]

ENCAPSULAMIENTO

Nombre en inglés: Encapsulation (Bansiya [31]), Encapsulated (Dromey [26])

También conocido como: Encapsulado (Dromey [26])

Descripción:

IEEE 610.12 [23]: Una técnica de desarrollo de software que consiste en aislar una función de sistema o un conjunto de datos y operaciones sobre esos datos dentro de un modulo y proveer especificaciones precisas para el módulo.

Otras definiciones:

Dromey [26]: La manera como son usadas las variables puede tener un impacto significativo sobre la modularidad y por lo tanto la calidad independiente de los módulos, programas y sistemas

Bansiya [31]: Definido como el adjuntar de los datos y comportamiento dentro de una construcción simple. En diseño Orientado a Objetos, la propiedad específicamente refiere a clases de diseño que previenen acceso a declaraciones de atributos que por su definición son privados, entonces protege la representación interna de los objetos

Atributos relacionados: Mantenibilidad, Confiabilidad, Portabilidad, Reusabilidad, Comprensibilidad, Efectividad

Modelos en los que aparece: Dromey [26], Bansiya [31]

COMPRESIBILIDAD

Nombre en inglés: Understandability

También conocido como: Comprensibilidad

Descripción:

Boehm [17]: Medida en que la finalidad del producto está clara para el evaluador.

ISO 9126 [15]: La capacidad del producto software para permitirle al usuario entender si el software es conveniente, cómo puede ser utilizado para las tareas determinadas y las condiciones de uso.

Otras definiciones:

Bansiya [31]: Las propiedades de un diseño que posibilita la facilidad de aprendizaje y comprensión. Esta directamente relacionado con la Complejidad de la estructura de diseño

QUINT2 [58] [59]: Esfuerzo de los usuarios para reconocer el concepto lógico y su aplicabilidad

JETTER [3]: El código es fácil de leer en el sentido que un inspector pueda rápidamente reconocer su propósito

MCCONNELL [66]: La facilidad con la que puede comprenderse un sistema en los niveles de organización del sistema y sentencias detalladas.

Atributos relacionados: Usabilidad, Encapsulamiento, Cohesión, Estructurado, auto descripción, concisión, legibilidad

Modelos en los que aparece: Boehm [17], ISO 9126 [15], Bansiya [31], QUINT2 [58] [59]

EXACTITUD

Nombre en inglés: Accuracy

También conocido como: Preciso (Dromey [26]), Veracidad

Descripción:

McCall [18]: Los atributos de software que proporcionan la precisión requerida en cálculos y salidas.

Boehm [17]: Medida en la que las salidas de un producto software son suficientemente precisas para satisfacer su uso deseado

Otras definiciones:

ISO 9126 [15]: La capacidad del producto de software para proporcionar resultados correctos o efectos convenidos con el grado necesario de precisión. (Esto incluye el grado necesario de precisión de los valores calculados).

QUINT2 [58] [59]: Provisión de resultados o efectos correctos o concertados

Dromey [26]: Una variable o constante es de tipo impreciso cuando su precisión no es suficiente para conocer la exactitud requerida para la computación.

SOFTWARE QUALITY ASSURANCE GROUP [65]: La característica del software que proporciona la precisión requerida en cálculos salidas.

PRESSMAN [1]: La exactitud de los cálculos y del control.

IEEE 610.12 [23]: (1) Una valoración cuantitativa de corrección, o libertad de error. (2) una medida cuantitativa de la magnitud del error.

MCCONNELL [66]: El grado en que un sistema, construido, es libre de error, especialmente con respecto a salidas cuantitativas.

Atributos relacionados: Confiabilidad, Desempeño, Funcionalidad

Modelos en los que aparece: McCall [18], Boehm [17], FURPS [20], ISO 9126 [15], Dromey [26], QUINT2 [58] [59]

FACILIDAD DE CAMBIO

Nombre en inglés: Changeability, Expandability (McCall [18]), Extendability (IEEE 610.12 [23]), Augmentability (Boehm [17]), Extendibility (Bansiya [31], FURPS [20])

También conocido como: Facilidad de cambio, Modificabilidad (Boehm [17]), Extensibilidad (Bansiya [31], FURPS [20]), Facilidad de cambio.

Descripción:

ISO 9126 [15]: La capacidad del producto software para permitir una modificación especificada que debe ser implementada

McCall [18]: Atributos de software que son proporcionados para expansión de requerimientos de almacenamiento de datos o funciones computacionales

Otras definiciones:

QUINT2 [58] [59]: Esfuerzo necesitado para modificación, remoción de fallos o para cambios del ambiente.

Boehm [17]: Medida en la que un producto software fácilmente es expandido adecuadamente en los requerimientos de almacenamiento de datos o componentes computacionales funcionales

Bansiya [31]: Refiere a la presencia y uso de las propiedades en un diseño existente de tal manera que permite para la incorporación de nuevos requerimientos en el diseño

SOFTWARE QUALITY ASSURANCE GROUP [65]: el esfuerzo requerido para incrementar capacidad o rendimiento del software incrementando funciones actualizadas o adicionando nuevas funciones o datos. Las características del software que proporcionan la capacidad de expansión para funciones o datos.

PRESSMAN [1]: El grado con que pueden ampliarse el diseño arquitectónico, de datos o procedimental

IEEE 610.12 [23]: La facilidad con que un sistema o componente puede ser modificado para incrementar almacenamiento o capacidad funcional.

BERANDER [36]: La capacidad de un producto software de permitir una modificación específica a ser implementada.

Atributos relacionados: Mantenibilidad, Flexibilidad, Abstracción, Herencia, Polimorfismo, Soportabilidad

Modelos en los que aparece: ISO 9126 [15], QUINT2 [58] [59], McCall [18], Boehm [17], FURPS [20], Bansiya [31]

HERENCIA

Nombre en inglés: Inheritance

Descripción:

Bansiya [31]: Una medida de relaciones "es-una" entre clases. Estas relaciones están relacionadas para el nivel de anidamiento de clases en una Herencia jerárquica

Atributos relacionados: Facilidad de cambio, Efectividad

Modelos en los que aparece: Bansiya [31]

INDEPENDENCIA DE MAQUINA

Nombre en inglés: Machine Independence, Device independence

También conocido como: independencia de dispositivo (Boehm [17]), independencia del hardware

Descripción:

McCall [18]: atributos de software que determinan su dependencia sobre un sistema hardware

Otras definiciones:

Boehm [17]: Medida en la que un producto software puede ser ejecutado sobre otras configuraciones de hardware de computador que una en uso en el mismo momento

PRESSMAN [1]: el grado con que es desacoplado el software del hardware donde opera
Atributos relacionados: Portabilidad, Reusabilidad
Modelos en los que aparece: McCall [18], Boehm [17].

INDEPENDENCIA DEL SISTEMA SOFTWARE

Nombre en inglés: Software system independence

Descripción:

McCall [18]: atributos de software que determinan su dependencia sobre un entorno software (sistemas operativos, utilidades, rutinas de E/S, etc.)

Otras definiciones:

ISO 9126 [15]: La capacidad del producto software para coexistir con otro software independiente en un ambiente común en el cual comparten recursos comunes

PRESSMAN [1]: El grado de independencia de programa respecto a las características del lenguaje de programación no estándar, características del sistema operativo y otras restricciones del entorno.

Atributos relacionados: Portabilidad, Reusabilidad

Modelos en los que aparece: McCall [18]

Otros: en ISO 9126 [15] es conocido como co-existencia

INSTALABILIDAD

Nombre en inglés: Installability

También conocido como: Facilidad de instalación

Descripción:

ISO 9126 [15]: La capacidad del producto software para ser instalado en un ambiente especificado

Otras definiciones:

QUINT2 [58] [59]: Esfuerzo necesitado para instalar el software en un ambiente específico

Atributos relacionados: Portabilidad, Soportabilidad

Modelos en los que aparece: FURPS [20], ISO 9126 [15], QUINT2 [58] [59]

MODULARIDAD

Nombre en inglés: Modularity

También conocido como: Estructurado (Boehm [17], Dromey [26])

Descripción:

McCall [18]: atributos de software que proporcionan una estructura de módulos altamente independientes

Otras definiciones:

SQAE [30]: ¿El código ha sido Estructurado dentro de segmentos controlables los cuales minimicen el acoplamiento bruto y simplifique el entendimiento?

Boehm [17]: Medida en la que un producto software posee y define patrones de organización de sus partes interdependientes

Dromey [26]: una forma estructural exhibe la propiedad de ser estructurada si sigue las reglas de programación estructurada

SOFTWARE QUALITY ASSURANCE GROUP [65]: las características del software que proporcionan una estructura de módulos altamente independientes

PRESSMAN [1]: La independencia funcional de componentes de programa

IEEE 610.12 [23]: el grado en que un sistema o programa de computador es compuesta por componentes discretos tal que el cambio de un componente tiene un impacto mínimo en otros componentes

Atributos relacionados: Mantenibilidad, Evolución, Portabilidad, flexibilidad, Testeabilidad, Reusabilidad, Interoperabilidad, Comprensibilidad, Modificabilidad, Funcionalidad, Confiabilidad

Modelos en los que aparece: McCall [18], SQAE [30]

POLIMORFISMO

Nombre en inglés: Polymorphism

También conocido como:

Descripción:

Bansiya [31]: La habilidad de sustituir objetos cuyas interfaces concuerdan para más de una en tiempo de ejecución. Esta es una medida de servicios que son dinámicamente determinados en tiempo de ejecución en un objeto

Atributos relacionados: Confiabilidad, Funcionalidad, Facilidad de cambio, Efectividad

Modelos en los que aparece: Bansiya [31]

TAMAÑO

Nombre en inglés: Size, design size

También conocido como: tamaño del diseño

Descripción:

IEEE 1045: Tamaño referido al número de directo y personal de apoyo involucrado en el proyecto. Registro del número de personas en el proyecto al máximo y promedio nivel de personal.

Otras definiciones:

Bansiya [31]: Una medida del número de clases usada en el diseño

Atributos relacionados: Reusabilidad, Funcionalidad

Modelos en los que aparece: Bansiya [31]

GENERALIDAD

Nombre en inglés: Generality

También conocido como: Genérico (Dromey [26])

Descripción:

McCall [18]: atributos de software que proporcionan extensión a las funciones realizadas.

Otras definiciones:

Dromey [26]: Un modulo es genérico si sus cálculos son abstraídos para una forma de tipo parametrizado

PRESSMAN [1]: la amplitud potencial de los componentes de un programa

SOFTWARE QUALITY ASSURANCE GROUP [65]: las características del software que proporcionan amplitud a las funciones realizadas.

IEEE 610.12 [23]: el grado en que un sistema o componente realiza un rango amplio de funciones.

Atributos relacionados: Flexibilidad, Reusabilidad, Funcionalidad

Modelos en los que aparece: McCall [18], Dromey [26], FURPS [20]

INGENIERÍA HUMANA

Nombre en inglés: human engineering, Human factor

También conocido como: Factor humano (FURPS [20])

Descripción:

Boehm [17]: Medida en la que un producto software cumple a cabalidad sus propósitos sin que los usuarios gasten su tiempo y energía o degraden su moral.

Atributos relacionados: accesibilidad, Comunicatividad, Usabilidad

Modelos en los que aparece: Boehm [17], FURPS [20]

SIMPLICIDAD

Nombre en inglés: Simplicity

También conocido como: Sencillez

Descripción:

McCall [18]: los atributos del software que proporcionan implementación de funciones en la manera más comprensible. (Usualmente evita prácticas con incremento de Complejidad)

Otras definiciones:

SOFTWARE QUALITY ASSURANCE GROUP [65]: las características del software que proporciona la definición e implementación de funciones en la menor Complejidad y la manera más entendible.

PRESSMAN [1]: El grado de Facilidad con que puede entenderse un programa

IEEE 610.12 [23]: el grado en que un sistema o componente tiene un diseño e implementación que es sincero y fácil de entender

Atributos relacionados: Confiabilidad, Mantenibilidad, Testeabilidad

Modelos en los que aparece: McCall [18]

TRAZABILIDAD

Nombre en inglés: Traceability

También conocido como: Rastreabilidad

Descripción:

McCall [18]: Los atributos del software que proporcionan un hilo desde los requisitos para la implementación en relación con el desarrollo específico y el entorno operativo

Otras definiciones:

QUINT2 [58] [59]: Facilidad para la correcta verificación de procesamiento de datos sobre puntos requeridos

SOFTWARE QUALITY ASSURANCE GROUP [65]: las características del software que proporcionan un hilo desde los requerimientos para la implementación con respecto al desarrollo específico y entorno operacional.

IEEE 610.12 [23]: (1) el grado en que una relación puede ser establecida entre dos o más productos del proceso de desarrollo, especialmente productos que tienen un predecesor o sucesor. (2) La característica de un sistema que permita la

identificación y control de relaciones entre requerimientos, componentes software, datos, y documentación a diferentes niveles en la jerarquía del sistema.

PRESSMAN [1]: La capacidad de seguir una representación del diseño o un componente real del programa hasta los requisitos.

Atributos relacionados: Corrección, Funcionalidad

Modelos en los que aparece: McCall [18], QUINT2 [58] [59]

AJUSTABLE

Nombre en inglés: Adjustable

Descripción:

Dromey [26]: Una forma estructural es ajustable si no contiene variables no declaradas (aparte de 1, 0, o -1) y si el número mínimo de variables de proposición-simple necesitadas para soportar los cálculos realizados son usados

Atributos relacionados: Mantenibilidad, Portabilidad, Reusabilidad,

Modelos en los que aparece: Dromey [26]

ASIGNADO

Nombre en inglés: Assigned

Descripción:

Dromey [26]: Una variable es asignada si esta recibe cualquier valor por asignar, entrada, o parámetro anterior asignado para su uso. La propiedad de asignado es una especialización de la propiedad de completitud que Aplica específicamente a variables y datos estructurales de todo tipo

Atributos relacionados: Funcionalidad, Confiabilidad

Modelos en los que aparece: Dromey [26]

Otros: Es una especialización de completitud

COMPUTABLE

Nombre en inglés: computable

Descripción:

Dromey [26]: Una forma estructural es computable si esta solo involucra cálculos que son definidos de acuerdo a una teoría estándar de computación y están dentro de los límites definidos por el programa, el lenguaje de programación y/o la maquina

Atributos relacionados: Funcionalidad, Confiabilidad

Modelos en los que aparece: Dromey [26]

CONTROL DE ANOMALÍAS

Nombre en inglés: Anomaly control

Descripción:

SQAE [30]: ¿tiene provisión para manejo comprensivo de error y procesamiento de excepciones detallados y aplicados?

Atributos relacionados: Mantenibilidad, Evolución

Modelos en los que aparece: SQAE [30]

DIRECTO

Nombre en inglés: Direct

Descripción:

Dromey [26]: Un cómputo es expresado directamente si la Abstracción, elección de la representación y la estructura del cómputo son congruentes con el problema original es modelado por la computación

Atributos relacionados: Mantenibilidad, Eficiencia.

Modelos en los que aparece: Dromey [26]

DISEÑO SIMPLE

Nombre en inglés:

Descripción:

SQAE [30]: ¿El código prestado a sí mismo para legibilidad y trazabilidad donde el comportamiento dinámico puede ser fácilmente predicho para análisis estático?

Atributos relacionados: Mantenibilidad, Evolución

Modelos en los que aparece: SQAE [30]

DOCUMENTADO

Nombre en inglés: Documented

También conocido como: Este atributo es asumido en DOCUMENTACIÓN

Descripción:

Dromey [26]: Una forma estructural es documentada si su propósito, estrategia, determinación y propiedades son todas explícita y precisamente definidas dentro del contexto de la forma estructural.

Atributos relacionados: Mantenibilidad, Usabilidad, Portabilidad, Reusabilidad

Modelos en los que aparece: Dromey [26]

ESPECIFICADO

Nombre en inglés: Specified

También conocido como: Este atributo es sumido en AUTO DESCRIPCIÓN

Descripción:

Dromey [26]: Un modulo, programa o forma estructural es especificado si sus Funcionalidades son descritas por precondiciones y pos condiciones

Atributos relacionados: Funcionalidad, Mantenibilidad, Confiabilidad, Usabilidad, Portabilidad, Reusabilidad

Modelos en los que aparece: Dromey [26]

HOLGADAMENTE ACOPLADO

Nombre en inglés: Loosely coupled

También conocido como:

Descripción:

Dromey [26]: Un modulo o programa esta holgadamente acoplado si todos las llamadas a módulos son datos-acoplados para la invocación a programa/modulo

Atributos relacionados: Mantenibilidad, Confiabilidad, Portabilidad, Reusabilidad

Modelos en los que aparece: Dromey [26]

INICIALIZADO

Nombre en inglés: Initialized

También conocido como:

Descripción:

Dromey [26]: Una estructura de lazo está inicializada si todas sus variables en un lazo son previamente inicializadas para entrar al lazo tan tarde como sea posible en prioridad para la entrada al lazo. La propiedad de inicializado es una especialización de la propiedad asignado que aplica para lazos.

Atributos relacionados: Funcionalidad, Mantenibilidad, Confiabilidad

Modelos en los que aparece: Dromey [26]

MENSAJES

Nombre en inglés: Messaging

También conocido como:

Descripción:

Bansiya [31]: Una cuenta del número de métodos públicos que están disponibles como servicios para otras clases. Esta es una medida de los servicios que las clases proveen

Atributos relacionados: Reusabilidad, Funcionalidad

Modelos en los que aparece: Bansiya [31]

NO REDUNDANTE

Nombre en inglés: Nonredundant

También conocido como:

Descripción:

Dromey [26]: Una forma estructural exhibe la propiedad de ser no redundante cuando tiene todos los elementos lógicos necesarios y solo los necesarios para definir la forma estructural

Atributos relacionados: Mantenibilidad, Eficiencia.

Modelos en los que aparece: Dromey [26]

PARAMETRIZADO

Nombre en inglés: Parameterized

También conocido como:

Descripción:

Dromey [26]: Un módulo es parametrizado si contiene como parámetros todas y solo las entradas y salidas suficientes y necesarias para caracterizar una función/procedimiento particular bien definido

Atributos relacionados: Mantenibilidad, Portabilidad, Reusabilidad

Modelos en los que aparece: Dromey [26]

PROGRESIVO

Nombre en inglés: Progressive

También conocido como:

Descripción:

Dromey [26]: Un lazo o algoritmo recursivo es progresivo si allí hay evidencia clara que la estructura haga progresar hacia la terminación con cada iteración o llamada recursiva y la función de variante de asociación está definida por debajo de cero

Atributos relacionados: Funcionalidad, Mantenibilidad, Confiabilidad

Modelos en los que aparece: Dromey [26]

RANGO INDEPENDIENTE

Nombre en inglés: Range Independent

También conocido como:

Descripción:

Dromey [26]: Una forma estructural es rango independiente si tanto sus bajadas y subidas delimitadas no son números fijos o constantes carácter. Esta propiedad es aplicada particularmente a especificaciones de array y estructuras iterativas designadas para procesar un segmento de elementos en un array.

Atributos relacionados: Mantenibilidad, Reusabilidad

Modelos en los que aparece: Dromey [26]

RESUELTO

Nombre en inglés:

También conocido como:

Descripción:

Dromey [26]: Cuando el control de la estructura de la implementación involucrada concuerda con la estructura de los datos o el problema en el sentido de Jackson (esto es, la estructura de control concuerda con la estructura de datos y por esta razón satisfaga el principio de correspondencia)

Atributos relacionados: Mantenibilidad, Eficiencia.

Modelos en los que aparece: Dromey [26]

UTILIZADO

Nombre en inglés: Utilized

También conocido como:

Descripción:

Dromey [26]: una forma estructural es utilizada si esta ha sido definida y además usada dentro de su alcance. Esta propiedad es aplicada a todas las formas de estructuras de datos y módulos

Atributos relacionados: Mantenibilidad, Eficiencia.

Modelos en los que aparece: Dromey [26]

VARIANTE

Nombre en inglés: Variant

También conocido como:

Descripción:

Dromey [26]: Un lazo seguro es variante si este define una relación que sea congruente con, y en forma derivable, la función variante usada para la terminación confirmada del lazo

Atributos relacionados: Funcionalidad, Mantenibilidad, Confiabilidad

Modelos en los que aparece: Dromey [26]

CAPA 3: FC-EXCLUSIVOS-BAJO NIVEL

ACOPLAMIENTO

Nombre en inglés: Coupling

Descripción:

Bansiya [31]: Define la interdependencia de un objeto sobre otros objetos en un diseño. Esta es una medida del número de otros objetos que podrían tener ser accedidos por un objeto en orden para ese objeto a función correctamente

Otras definiciones:

IEEE 610.12 [23]: la forma y grado de independencia entre módulos software.

Atributos relacionados: Cohesión

Modelos en los que aparece: Bansiya [31]

AMIGABILIDAD

Nombre en inglés: User friendliness

Descripción:

QUINT2 [58] [59]: Satisfacción del usuario

Atributos relacionados: Usabilidad

Modelos en los que aparece: QUINT2 [58] [59]

ANALIZABILIDAD

Nombre en inglés: Analyzability

Descripción:

ISO 9126 [15]: La capacidad del producto software para ser diagnosticado por deficiencias o causas de fallas en el software, o para identificar las partes que deben modificarse

Otras definiciones:

QUINT2 [58] [59]: Esfuerzo necesitado para diagnóstico de deficiencias o causas de fallos, o para identificación de partes a ser modificadas.

BERANDER [36]: la capacidad del producto software de ser diagnosticado por deficiencias o causas de fallas en el software, o para identificar las partes a ser modificadas.

Atributos relacionados: Mantenibilidad

Modelos en los que aparece: ISO 9126 [15], QUINT2 [58] [59]

ASISTENCIA DE USUARIO

Nombre en inglés: Serviceability, Helpfulness, Training

También conocido como: Entrenamiento o Formación (McCall [18])

Descripción:

QUINT2 [58] [59]: Habilidad de instrucción para el usuario sobre cómo interactuar con el software

Otras definiciones:

SOFTWARE QUALITY ASSURANCE GROUP [65]: las características del software que proporcionan transmisión desde una operación actual y provee familiarización inicial.

Atributos relacionados: Usabilidad

Modelos en los que aparece: FURPS [20], QUINT2 [58] [59]

ATRATIVO

Nombre en inglés: Attractiveness, Aesthetics

También conocido como: Estética (FURPS [20])

Descripción:

ISO 9126 [15]: La capacidad del producto software para que sea agradable al usuario. Esto refiere a los atributos del software previstos para hacer el software más atractivo al usuario, tal como el uso del color y de la naturaleza del diseño gráfico

Otras definiciones:

QUINT2 [58] [59]: Satisfacción de los deseos y preferencias de los usuarios latentes, a través de servicios, conductas y presentación más allá de las demandas actuales

Atributos relacionados: Usabilidad

Modelos en los que aparece: ISO 9126 [15], QUINT2 [58] [59], FURPS [20]

COMPLEJIDAD

Nombre en inglés: Complexity

Descripción:

IEEE 610.12 [23]: (1) el grado en que un sistema o componente tiene un diseño o implementación que es difícil de entender y verificar.

Otras definiciones:

Bansiya [31]: Una medida del grado de dificultad en entendimiento y Comprensibilidad de estructuras internas y externas de clases y sus relaciones.

Atributos relacionados: Comprensibilidad

Modelos en los que aparece: Bansiya [31]

COMPORTAMIENTO EN EL TIEMPO

Nombre en inglés: Time behaviour

Descripción:

ISO 9126 [15]: La capacidad del producto software para proporcionar adecuados tiempos de respuesta, de procesamiento y de tasas de Eficiencia en el Desempeño de su función, bajo condiciones establecidas

Otras definiciones:

QUINT2 [58] [59]: Tiempos de respuesta y procesamiento, e índices de rendimiento

Atributos relacionados: Eficiencia

Modelos en los que aparece: ISO 9126 [15], QUINT2 [58] [59]

ESTABILIDAD

Nombre en inglés: Stability

También conocido como:

Descripción:

ISO 9126 [15]: La capacidad del producto software para evitar efectos no esperados debido a modificaciones en el software

Otras definiciones:

QUINT2 [58] [59]: Riesgo de efecto inesperado por modificaciones

BERANDER [36]: la capacidad de un producto software de evitar efectos inesperados desde modificaciones del software.

Atributos relacionados: Mantenibilidad

Modelos en los que aparece: ISO 9126 [15], QUINT2 [58] [59]

FACILIDAD DE APRENDIZAJE

Nombre en inglés: Learnability

También conocido como:

Descripción:

ISO 9126 [15]: La capacidad del producto software para permitirle al usuario aprender su aplicación (como por ejemplo, control de operación, entradas, salidas).

Otras definiciones:

QUINT2 [58] [59]: Esfuerzo de los usuarios para aprender las aplicaciones del software (por ejemplo control, entrada y salida)

Atributos relacionados: Usabilidad

Modelos en los que aparece: ISO 9126 [15], QUINT2 [58] [59]

IDONEIDAD

Nombre en inglés: Suitability

También conocido como: Apropiabilidad, aplicabilidad.

Descripción:

ISO 9126 [15]: La capacidad del producto software para proporcionar un apropiado conjunto de funciones para las tareas especificadas y los objetivos de los usuarios

Otras definiciones:

QUINT2 [58] [59]: Presencia y propiedad de una serie de funciones para tareas específicas.

Atributos relacionados: Funcionalidad

Modelos en los que aparece: ISO 9126 [15], QUINT2 [58] [59]

JERARQUÍA

Nombre en inglés: Hierarchies

Descripción:

Bansiya [31]: las jerarquías son usadas para representar diferentes conceptos de Generalizaciones-Especializaciones en un diseño. Es una cuenta del número no ancestral de clases que tienen hijos en el diseño

Atributos relacionados: Funcionalidad

Modelos en los que aparece: Bansiya [31]

OPERATIVIDAD

Nombre en inglés: Operability

También conocido como: Facilidad de operación

Descripción:

McCall [18]: atributos de software que determinan operación y procedimientos concernientes con la operación del software

Otras definiciones:

ISO 9126 [15]: Esfuerzo de los usuarios para operación y control de operación

Atributos relacionados: Usabilidad

QUINT2 [58] [59]: La capacidad del producto software para permitirle al usuario su operación y control

SOFTWARE QUALITY ASSURANCE GROUP [65]: la característica del software que determina operaciones y procedimientos concernientes con operaciones del software y que proporciona entradas usables y salidas que pueden ser asimiladas.

PRESSMAN [1]: Facilidad de operación de un programa

Atributos relacionados: Usabilidad

Modelos en los que aparece: McCall [18], ISO 9126 [15], QUINT2 [58] [59]

PERSONALIZACIÓN

Nombre en inglés: Customizability

Descripción:

QUINT2 [58] [59]: Habilidad de ser personalizado por el usuario para reducir el esfuerzo requerido para uso e incrementar la satisfacción con el software

Otras definiciones:

ALVARO [64]: este atributo mide el número de parámetros personalizables que el componente ofrece

Atributos relacionados: Usabilidad

Modelos en los que aparece: QUINT2 [58] [59]

RECUPERABILIDAD

Nombre en inglés: Recoverability

También conocido como: tiempo de recuperación

Descripción:

ISO 9126 [15]: La capacidad del producto software para restablecer su nivel de Desempeño y recuperar los datos directamente afectados en caso de falla (así como con el tiempo y el esfuerzo necesario para ello). Después de una falla, un producto de software algunas veces será dado de baja por cierto período de tiempo, esta longitud de tiempo es evaluada para su recuperabilidad

Otras definiciones:

QUINT2 [58] [59]: Capacidad de restablecer el nivel de Desempeño y recuperar los datos directamente afectados en caso de una falla, y el tiempo y esfuerzo requeridos para esto

Atributos relacionados: Confiabilidad

Modelos en los que aparece: FURPS [20], ISO 9126 [15], QUINT2 [58] [59]

SEGURIDAD

Nombre en inglés: security

Descripción:

ISO 9126 [15]: La capacidad del producto software para proteger la información y los datos de modo que las personas o los sistemas no autorizados no puedan leerlos o modificarlos, y a las personas o los sistemas autorizados no les niegue el acceso a ellos.

Otras definiciones:

QUINT2 [58] [59]: Habilidad para prevenir acceso no autorizado, tanto accidental como deliberado, a programas o datos.

PRESSMAN [1]: Consiste en la disponibilidad de mecanismos que controlen o protejan los programas o datos.

PENICHER [60]-CALERO: atributos de software que soportan la habilidad de prevenir acceso no autorizado, sea accidental o deliberado, a programas o datos.

Atributos relacionados: Funcionalidad

Modelos en los que aparece: FURPS [20], ISO 9126 [15], QUINT2 [58] [59]

TOLERANCIA AL ERROR

Nombre en inglés: Error tolerance, Fault tolerance

También conocido como: tolerancia a fallos

Descripción:

McCall [18]: los atributos del software que proporcionan continuidad de operación bajo condiciones no nominales

Otras definiciones:

ISO 9126 [15]: La capacidad del producto software para mantener un nivel de Desempeño especificado en casos de fallas (en el software) o de que infrinjan sus enlaces especificados. (El nivel de Desempeño especificado puede incluir la capacidad de falla segura).

QUINT2 [58] [59]: Habilidad para mantener un nivel especificado de Desempeño en caso de fallas de software o infracciones para su interface específica

PRESSMAN [1]: El daño causado cuando un programa encuentra un error

IEEE 610.12 [23]: (1) la habilidad de un sistema software o componente de continuar operación normal a pesar de la presencia de fallas hardware o software. (2) el número de fallas de un sistema o componente puede resistir antes del deterioro de operación normal. (3) relativo al estudio de errores, fallas, fallas y anomalías, y de métodos para permitir a los sistemas continuar la operación normal en la presencia de fallas.

Atributos relacionados: Confiabilidad.

Modelos en los que aparece: McCall [18], ISO 9126 [15], QUINT2 [58] [59]

CLARIDAD

Nombre en inglés: Clarity

Descripción:

QUINT2 [58] [59]: Claridad de la construcción de usuario experto de las funciones que el software puede desempeñar

Atributos relacionados: Usabilidad

Modelos en los que aparece: QUINT2 [58] [59]

INSTRUMENTACIÓN

Nombre en inglés: Instrumentation

Descripción:

McCall [18]: atributos de software que proporcionan para la medición de uso o identificación de errores.

Otras definiciones:

PRESSMAN [1]: El grado con que el programa vigila su propio funcionamiento e identifica los errores que ocurren

IEEE 610.12 [23]: dispositivos o instrucciones instalados o insertados en el hardware o software para monitorear la operación de un sistema o componente.

Atributos relacionados: Testeabilidad

Modelos en los que aparece: McCall [18]

LEGIBILIDAD

Nombre en inglés: legibility, readability (MCCONNELL [66])

Descripción:

Boehm [17]: Medida en la que las funciones de un producto software y aquellos de sus expresiones componentes son discernidas fácilmente por lectura al código

Otras definiciones:

MCCONNELL [66]: la Facilidad con que puede leerse y entenderse el código fuente de un sistema, especialmente en un nivel de declaración detallado.

Atributos relacionados: Comprensibilidad

Modelos en los que aparece: Boehm [17]

MADUREZ

Nombre en inglés: Maturity

Descripción:

ISO 9126 [15]: La capacidad del producto software para evitar fallas como resultado de fallas en el software.

Otras definiciones:

QUINT2 [58] [59]: Frecuencia de anomalía por fallas en el software

Atributos relacionados: Confiabilidad

Modelos en los que aparece: ISO 9126 [15], QUINT2 [58] [59]

REEMPLAZABILIDAD

Nombre en inglés: Replaceability

Descripción:

ISO 9126 [15]: La capacidad del producto software para ser utilizado en lugar de otro producto de software especificado para el mismo propósito en el mismo ambiente

Otras definiciones:

QUINT2 [58] [59]: Oportunidad y esfuerzo de utilización del software en el lugar de otro software específico en el ambiente de ese software

Atributos relacionados: Portabilidad

Modelos en los que aparece: ISO 9126 [15], QUINT2 [58] [59]

CAPACIDADES

Nombre en inglés: capabilities

Atributos relacionados: Funcionalidad

Modelos en los que aparece: FURPS [20]

CARACTERÍSTICAS

Nombre en inglés: Feature sets

Atributos relacionados: Funcionalidad

Modelos en los que aparece: FURPS [20]

COMPATIBILIDAD

Nombre en inglés: Compatibility

Nombre en inglés: Soportabilidad

Descripción:

IEEE 610.12 [23]: (1) la habilidad de dos o más sistemas o componentes de realizar sus funciones requeridas mientras comparten el mismo entorno hardware o software. (2) la habilidad de dos o más sistemas o componentes de intercambiar información.

Atributos relacionados: Soportabilidad

Modelos en los que aparece: FURPS [20]

Otros: Este atributo esta relacionado con los atributos de independencia de maquina e Interoperabilidad.

CONFIGURABILIDAD

Nombre en inglés: Configurability

Atributos relacionados: Soportabilidad

Modelos en los que aparece: FURPS [20]

Otros: relacionado con independencia de sistema software

DEGRADABILIDAD

Nombre en inglés: Degradability

Descripción:

QUINT2 [58] [59]: Esfuerzo necesitado para restablecer el funcionamiento esencial después de un breakpoint

Atributos relacionados: Confiabilidad

Modelos en los que aparece: QUINT2 [58] [59]

Otros: relacionado con recuperabilidad

EFICIENCIA DE ALMACENAMIENTO

Nombre en inglés: Storage efficiency

También conocido como:

Descripción:

McCall [18]: Los atributos de software que proporcionan para en los requerimientos de almacenamiento mínimos durante la operación

Otras definiciones:

IEEE 610.12 [23]: el grado en que un sistema o componente software realiza sus funciones designadas con un mínimo consumo de almacenamiento disponible

Atributos relacionados: Eficiencia

Modelos en los que aparece: McCall [18]

EFICIENCIA DE DISPOSITIVO

Nombre en inglés: device efficiency

Atributos relacionados: Eficiencia

Modelos en los que aparece: Boehm [17]

EFICIENCIA DE EJECUCIÓN

Nombre en inglés: Execution efficiency

Descripción:

McCall [18]: Los atributos de software que proporcionan durante un tiempo de procesamiento mínimo

Otras definiciones:

PRESSMAN [1]: El rendimiento del funcionamiento de un programa.

IEEE 610.12 [23]: el grado en que un sistema o componente realiza sus funciones designadas con un mínimo consumo de tiempo.

Atributos relacionados: Eficiencia

Modelos en los que aparece: McCall [18]

ESTANDARIZACIÓN DE LAS COMUNICACIONES

Nombre en inglés: Communications commonality

También conocido como: este atributo asume en CONFORMIDAD

Descripción:

McCall [18]: atributos de software que proporcionan el uso de protocolos estándar y rutinas de interface.

Otras definiciones:

PRESSMAN [1]: El grado de empleo de estándares de interfaces, protocolos y anchos de banda.

Atributos relacionados: Interoperabilidad

Modelos en los que aparece: McCall [18]

ESTANDARIZACIÓN DE LOS DATOS

Nombre en inglés: Data Commonality

También conocido como: este atributo asume en CONFORMIDAD

Descripción:

McCall [18]: atributos de software que proporcionan el uso del estándar de representaciones de datos

Otras definiciones:

PRESSMAN [1]: El empleo de estructuras y tipos de datos estándares a lo largo del programa
Atributos relacionados: Interoperabilidad
Modelos en los que aparece: McCall [18]

EVOLUCIÓN

Nombre en inglés: Evolution
Atributos relacionados: modularidad, auto descripción, diseño simple, control de anomalías, documentación
Modelos en los que aparece: SQA [30]

EXPLICITUD

Nombre en inglés: Explicitness
Descripción:
QUINT2 [58] [59]: Claridad del estatus del producto software (barras de progresión, etc.)
Atributos relacionados: Usabilidad
Modelos en los que aparece: SQA [30]

FRECUENCIA Y SEVERIDAD DE FALLAS

Nombre en inglés: Frequency and severity of failure
Atributos relacionados: Confiabilidad
Modelos en los que aparece: FURPS [20]
Otros: existe una métrica con el mismo nombre así que es posible que este no sea precisamente un atributo.

HOMOGÉNEO

Nombre en inglés:
Descripción:
Dromey [26]: Una forma iterativa o recursiva si esta puede ser descrita por una invariante donde el predicado mayor asume la forma conjuntiva.
Atributos relacionados: Mantenibilidad
Modelos en los que aparece: Dromey [26]

INDEPENDENCIA

Nombre en inglés: Independence
Descripción:
SQA [30]: ¿Tiene ataduras para especificar sistemas, extensiones, etc., minimizadas para facilitar eventual migración, Evolución, y/o incremento del proyecto?
Atributos relacionados: Portabilidad
Modelos en los que aparece: SQA [30]

INTERNALIZACIÓN

Nombre en inglés: Localizability
También conocido como: Localizabilidad

Atributos relacionados: Soportabilidad
Modelos en los que aparece: FURPS [20]

MANEJABILIDAD

Nombre en inglés: Manegeability

Descripción:

QUINT2 [58] [59]: Esfuerzo necesitado para (re)establecer el estatus de ejecución del software

Atributos relacionados: Mantenibilidad

Modelos en los que aparece: QUINT2 [58] [59]

PREDICTIBILIDAD

Nombre en inglés: Predictability

Atributos relacionados: Confiabilidad

Modelos en los que aparece: FURPS [20]

RENDIMIENTO

Nombre en inglés: throughput

Descripción:

IEEE 610.12 [23]: la cantidad de trabajado que puede realizar un sistema de computador o componente en un periodo determinado de tiempo; por ejemplo, número de trabajos por día.

Atributos relacionados: Desempeño

Modelos en los que aparece: FURPS [20]

TIEMPO MEDIO ENTRE FALLOS

Nombre en inglés: Mean time between failures

Descripción:

IEEE 610.12 [23]: el tiempo esperado u observado entre fallas consecutivas en un sistema o componente.

Atributos relacionados: Confiabilidad

Modelos en los que aparece: FURPS [20]

Otros: existe una métrica con el mismo nombre así que es posible que este no sea precisamente un atributo.

TIEMPO DE RESPUESTA

Nombre en inglés: response time

Descripción:

IEEE 610.12 [23]: el tiempo transcurrido entre el final de una investigación o comando a un sistema de computador interactivo y el comienzo de la respuesta del sistema.

Atributos relacionados: Desempeño

Modelos en los que aparece: FURPS [20]

VELOCIDAD

Nombre en inglés: Speed

Atributos relacionados: Desempeño

Modelos en los que aparece: FURPS [20]

ATRIBUTOS NO INCLUIDOS EN LAS CAPAS

ABSTRACCIÓN

Nombre en inglés: Abstraction

También conocido como: Este atributo ha sido asumido en ABSTRACTO

Descripción:

Bansiya [31]: Una medida aspectos de Generalización - especialización en el diseño. Las clases en el diseño que tienen uno o más descendientes exhibidos en esta propiedad de Abstracción

Atributos relacionados: Facilidad de cambio

Modelos en los que aparece: Bansiya [31]

CONSISTENTE

Nombre en inglés: Consistent

También conocido como: Este atributo es asumido en CONSISTENCIA

Descripción:

Dromey [26]: Una forma estructura es consistente si su uso mantiene sus propiedades o Funcionalidad y todos sus elementos contribuyen, a y refuerzan sus objetivos conjuntos o efectos

Atributos relacionados: Funcionalidad, Mantenibilidad, Confiabilidad, Usabilidad, Portabilidad, Reusabilidad

AYUDA

Nombre en inglés: Helpfulness

También conocido como: Este atributo es asumido en ASISTENCIA DE USUARIO

Descripción:

QUINT2 [58] [59]: Habilidad de instrucción para el usuario sobre como interactuar con el software

Atributos relacionados: Usabilidad

Modelos en los que aparece: Quint 2

EXPANSIBILIDAD

Nombre en inglés: Expandability, extendability (IEEE 610.12 [23]), Augmentability (Boehm [17]), extendibility (Bansiya [31], FURPS [20])

También conocido como: Facilidad de ser ampliado, Capacidad de crecimiento o incremento (Boehm [17]), capacidad de expansión, Extensibilidad.

Descripción:

McCall [18]: atributos de software que proporcionan para expansión de requerimientos de almacenamiento de datos o funciones computacionales

Boehm [17]: Medida en la que un producto software expande adecuadamente en los requerimientos de almacenamiento de datos o componentes computacionales funcionales

Otras definiciones:

Bansiya [31]: Refiere a la presencia y uso de las propiedades en un diseño existente de tal manera que permite para la incorporación de nuevos requerimientos en el diseño

SOFTWARE QUALITY ASSURANCE GROUP [65]: el esfuerzo requerido para incrementar capacidad o rendimiento del software incrementando funciones actualizadas o adicionando nuevas funciones o datos. Las características del software que proporcionan la capacidad de expansión para funciones o datos.

PRESSMAN [1]: El grado con que pueden ampliarse el diseño arquitectónico, de datos o procedimental

IEEE 610.12 [23]: La Facilidad con que un sistema o componente puede ser modificado para incrementar su almacenamiento o capacidad funcional.

Atributos relacionados: Flexibilidad, Abstracción, Herencia, Polimorfismo, Soportabilidad

Modelos en los que aparece: McCall [18], Boehm [17], FURPS [20], Bansiya [31]

TOLERANCIA A FALLOS

Nombre en inglés: Fault tolerance

También conocido como: este atributo es asumido como TOLERANCIA AL ERROR

Descripción:

ISO 9126 [15]: La capacidad del producto software para mantener un nivel de desempeño especificado en casos de fallas (en el software) o de que infrinjan sus enlaces especificados. (El nivel de desempeño especificado puede incluir la capacidad de falla segura).

Otras definiciones:

QUINT2 [58] [59]: Habilidad para mantener un nivel especificado de desempeño en caso de fallas de software o infracciones para su interface específica

Atributos relacionados: Confiabilidad

Modelos en los que aparece: ISO 9126 [15], QUINT2 [58] [59]

EFFECTIVIDAD

Nombre en inglés: Effectiveness

También conocido como: Este atributo ha sido asumido en EFICIENCIA

Descripción:

Bansiya [31]: Este refiere a la habilidad del diseño para conseguir la funcionalidad y comportamiento deseados usando conceptos de diseño Orientado a Objetos y técnicas

Atributos relacionados: Abstracción, Encapsulamiento, Composición, Herencia, Polimorfismo

EFFECTIVO

Nombre en inglés: Effective

También conocido como: Este atributo ha sido asumido en EFICIENCIA

Descripción:

Dromey [26]: Una forma estructural exhibe la propiedad de ser efectiva cuando tiene todos los elementos necesarios y solo los elementos necesarios para definir e implementar la forma estructural

Atributos relacionados: Mantenibilidad, Usabilidad, Eficiencia.

Modelos en los que aparece: Dromey [26]

ESTRUCTURADO

Nombre en inglés: Structuredness

También conocido como: Este atributo es asumido por MODULARIDAD

Descripción:

Boehm [17]: Medida en la que un producto software posee y define patrones de organización de sus partes interdependientes

Otras definiciones:

Dromey [26]: una forma estructural exhibe la propiedad de ser estructurada si sigue las reglas de programación estructurada

Atributos relacionados: Testeabilidad, Comprensibilidad, Modificabilidad, Funcionalidad, Mantenibilidad, Confiabilidad

Modelos en los que aparece: Boehm [17], Dromey [26]

EXTENSIBILIDAD

Nombre en inglés: Extensibility

También conocido como: Este atributo es asumido como FACILIDAD DE CAMBIO

Descripción:

Bansiya [31]: Refiere a la presencia y uso de las propiedades en un diseño existente de tal manera que permite para la incorporación de nuevos requerimientos en el diseño

Atributos relacionados: Soportabilidad, Abstracción, Herencia, Polimorfismo.

Modelos en los que aparece: FURPS [20], Bansiya [31]

Otros: es el mismo expandability según IEEE 610.12 [23]

GENÉRICO

Nombre en inglés: Generic

También conocido como: Este atributo es asumido como GENERALIDAD

Descripción:

Dromey [26]: Un modulo es genérico si sus cálculos son abstraídos para una forma de tipo parametrizado

Atributos relacionados: Mantenibilidad, Reusabilidad, Portabilidad

Modelos en los que aparece: Dromey [26]

MODIFICABILIDAD

Nombre en inglés: Modifiability

También conocido como: Este atributo es asumido como FACILIDAD DE CAMBIO

Descripción:

Boehm [17]: Medida en la que un producto software facilita la incorporación de cambios, en cuanto a la naturaleza de los cambios deseados haya sido determinada

Atributos relacionados: Estructurado, Capacidad de crecimiento, Mantenibilidad

Modelos en los que aparece: Boehm [17].

PRECISO

Nombre en inglés: Precise

También conocido como: Este atributo ha sido asumido por EXACTITUD

Descripción:

Dromey [26]: Una variable o constante es de tipo impreciso cuando su precisión no es suficiente para conocer la exactitud requerida para la computación

Atributos relacionados: Funcionalidad, Confiabilidad

Modelos en los que aparece: Dromey [26]

COHESIVO

Nombre en inglés: Cohesive

También conocido como: Este atributo es asumido en COHESIÓN

Descripción:

Dromey [26]: Una forma estructural es cohesiva si todos sus elementos son delimitados apretadamente uno al otro y todos ellos contribuyen a conseguir un objetivo simple o función.

Atributos relacionados: Mantenibilidad, Reusabilidad, Portabilidad

Modelos en los que aparece: Dromey [26]

CAPACIDAD DE CRECIMIENTO

Nombre en inglés: Augmentability

También conocido como: Este atributo ha sido asumido en FACILIDAD DE CAMBIO

Descripción:

Boehm [17]: Medida en la que un producto software expande adecuadamente en los requerimientos de almacenamiento de datos o componentes computacionales funcionales

Atributos relacionados: Funcionalidad

Modelos en los que aparece: Boehm [17]

CO-EXISTENCIA

Nombre en inglés: co-existence

También conocido como: Atributo es asumido en INDEPENDENCIA DE SISTEMA SOFTWARE

Descripción:

ISO 9126 [15]: La capacidad del producto software para coexistir con otro software independiente en un ambiente común en el cual comparten recursos comunes.

Atributos relacionados: Portabilidad

Modelos en los que aparece: ISO 9126 [15]

CONFORMIDAD CON LA CONFIABILIDAD

Nombre en inglés: Reliability Compliance

También conocido como: Este atributo es asumido en CONFORMIDAD

Descripción:

ISO 9126 [15]: La capacidad del producto software para adherirse a estándares, convenciones o regulaciones relacionadas con la Confiabilidad.

Atributos relacionados: Confiabilidad

Modelos en los que aparece: ISO 9126 [15]

CONFORMIDAD CON LA EFICIENCIA

Nombre en inglés: Efficiency Compliance

También conocido como: Este atributo es asumido en CONFORMIDAD

Descripción:

ISO 9126 [15]: La capacidad del producto software para adherirse a estándares o convenciones relacionadas con la Eficiencia

Atributos relacionados: Eficiencia

Modelos en los que aparece: ISO 9126 [15]

CONFORMIDAD CON LA FUNCIONALIDAD

Nombre en inglés: Functionality Compliance

También conocido como: Este atributo es asumido en CONFORMIDAD

Descripción:

ISO 9126 [15]: La capacidad del producto software para adherirse a estándares, normas, convenciones o regulaciones en legislaciones y prescripciones similares relacionadas con la Funcionalidad

Atributos relacionados: Funcionalidad

Modelos en los que aparece: ISO 9126 [15]

CONFORMIDAD CON LA MANTENIBILIDAD

Nombre en inglés: Maintainability Compliance

También conocido como: Este atributo es asumido en CONFORMIDAD

Descripción:

ISO 9126 [15]: La capacidad del producto software para adherirse a estándares o convenciones relacionadas con la Facilidad de Mantenimiento (Mantenibilidad)

Atributos relacionados: Mantenibilidad

Modelos en los que aparece: ISO 9126 [15]

CONFORMIDAD CON LA PORTABILIDAD

Nombre en inglés: Portability Compliance

También conocido como: Este atributo es asumido en CONFORMIDAD

Descripción:

ISO 9126 [15]: La capacidad del producto software para adherirse a estándares o convenciones relativas a la Portabilidad

Atributos relacionados: Portabilidad

Modelos en los que aparece: ISO 9126 [15]

CONFORMIDAD CON LA USABILIDAD

Nombre en inglés: Usability Compliance

También conocido como: Este atributo es asumido en CONFORMIDAD

Descripción:

ISO 9126 [15]: La capacidad del producto software para adherirse a estándares, convenciones, guías de estilo o regulaciones relacionadas con la Facilidad de uso.

Atributos relacionados: Usabilidad

Modelos en los que aparece: ISO 9126 [15]

CONTROL DE ACCESO

Nombre en inglés: Access control

También conocido como: Este atributo es asumido en ACCESIBILIDAD

Descripción:

McCall [18]: Atributos de software que proporcionan para control de acceso de software y datos

Atributos relacionados: Integridad

Modelos en los que aparece: McCall [18]

ENTRENAMIENTO

Nombre en inglés: Training

También conocido como: Formación. Este atributo es asumido como ASISTENCIA DE USUARIO

Descripción:

McCall [18]: Atributos de software que proporcionan transición desde operaciones actuales o familiarización inicial.

Otras definiciones:

PRESSMAN [1]: *Formación.* El grado en que ayuda el software a manejar el sistema a los nuevos usuarios

IEEE 1045: Entrenamiento refiere a entrenamiento profesional formal. Esto incluye cursos de formación externa, como seminarios, cursos de formación en la casa, tal como entrenamiento de estándares de código de la compañía, software, entorno de desarrollo, técnicas de prueba, etc. Registra el número promedio de días que cada empleado de proyecto gasta en entrenamiento por año (por ejemplo, un empleado recibe un promedio de 6.5 días de entrenamiento por año)

Atributos relacionados: Eficiencia

Modelos en los que aparece: McCall [18]

ESTÉTICA

Nombre en inglés: Aesthetics

También conocido como: atributo es asumido en ATRACTIVO

Atributos relacionados: Usabilidad

Modelos en los que aparece: FURPS [20]

FACILIDAD DE AUDITORIA

Nombre en inglés: Access audit

También conocido como: Auditoria de acceso, Este atributo es asumido en CONFORMIDAD

Descripción:

McCall [18]: Atributos de software que proporcionan para una auditoria del acceso de software y datos

Otras definiciones:

PRESSMAN [1]: La facilidad con la que puede comprobarse el cumplimiento de los estándares.

IEEE 610.12 [23]: Una exanimación independiente de un producto de trabajo o conjunto de productos de trabajo para asegurar conformidad con especificaciones, estándares, contrato contractual, u otros criterios.

Atributos relacionados: Eficiencia

Modelos en los que aparece: McCall [18]

FACTOR HUMANO

Nombre en inglés: Human factor

También conocido como: Este atributo es asumido como INGENIERÍA HUMANA

Atributos relacionados: Usabilidad

Modelos en los que aparece: FURPS [20]

UTILIZACIÓN DE RECURSOS

Nombre en inglés: Resource utilization

También conocido como: Este atributo es asumido en CONSUMO DE RECURSOS

Descripción:

ISO 9126 [15]: La capacidad del producto software para utilizar una apropiada cantidad y tipos de recursos cuando el software desempeña su función bajo condiciones establecidas

Atributos relacionados: Eficiencia

Modelos en los que aparece: ISO 9126 [15]

CAPACIDAD DE SERVICIO

Nombre en inglés: serviceability,

También conocido como: Este atributo ha sido asumido en ASISTENCIA DE USUARIO

Descripción:

Cualidad de ser capaz de proveer un buen servicio. [84]

Atributos relacionados: So Portabilidad

Modelos en los que aparece: FURPS [20]

ENCAPSULADO

Nombre en inglés: Encapsulated

También conocido como: Este atributo ha sido asumido en ENCAPSULAMIENTO

Descripción:

Dromey [26]: La manera como son usadas las variables puede tener un impacto significativo sobre la modularidad y por lo tanto la calidad independiente de los módulos, programas y sistemas

Atributos relacionados: Mantenibilidad, Reusabilidad, Portabilidad, Reusabilidad

Modelos en los que aparece: Dromey [26]

Los atributos han sido ubicados en sus respectivas capas y espectro, este ultimo relacionado a los criterios de calidad dados en el capítulo 2 y teniendo en cuenta los resultados del Anexo D. además son colocados la sección de atributos que no han sido incluidos, esto en consideración que ya han sido asumidos dentro de otros atributos que si están en las capas propuestas en el capítulo 3.

ANEXO H: LAS MÉTRICAS DE CALIDAD ENCONTRADAS

Éste anexo presenta las métricas encontradas en el transcurso de este proyecto:

Los términos usados para las métricas son los siguientes:

Nombre en español: Indica el nombre en español dado para la métrica

Nombre en inglés: Indica el nombre en español dado para la métrica

Descripción: Da una breve descripción de la métrica en forma de prosa

Fórmula: En el caso de existir fórmula para la métrica es enunciado en este apartado

Atributo al que ayuda a calcular: El atributo que calcula o mide.

Influencia sobre otros atributos: Si los hay, enuncia atributos sobre los cuales la métrica puede tener influencia ya sea directa o indirecta.

Granularidad: Indica en qué nivel para aplicar la métrica. Puede ser: clase, sistema, programa, modulo, paquete, método o variable.

Rango: en caso de que exista refiere al rango de valores en los cuales la métrica opera.

Referencias: Refiere a los autores que mencionan esta métrica. (Solo son colocadas las referencias más relevantes)

Espectro: En qué clasificación está la métrica después de haber aplicado los criterios de selección (Ver capítulo 2), puede ser, Claro, Gris y Oscuro

Otros: Menciona otras características de la métrica no consideradas

Como es posible observar, la clasificación indica que la métrica esta en un nivel más alto según su color (espectro), el *claro* (color verde) son aquellas cuya información está más completa y son más sencillas de entender; el *gris* (color rojizo) indica un nivel intermedio, es decir alguno de los criterios no cumplió a cabalidad, el *oscuro* (color naranja) indica que la métrica está incompleta, esto dificulta su estudio.

En el encabezado de cada métrica es colocado el color del espectro para el cual ha sido clasificado en el anexo D. A continuación están presentes las métricas encontradas:

MÉTRICA 1

Nombre en español: Medida de integridad

Descripción: Usa dos conceptos, Amenaza: Es la probabilidad de que un ataque de un tipo determinado ocurra en un tiempo determinado, Seguridad: es la probabilidad de que pueda repeler un ataque de un tipo determinado.

Fórmula:

$$\text{Integridad} = \sum [(1-\text{Amenaza}) \cdot (1-\text{Seguridad})], \text{ para cada tipo de ataque}$$

Atributo al que ayuda a calcular: Integridad

Granularidad: Sistema

Referencias: PRESSMAN [1]
Espectro: Gris

MÉTRICA 2

Nombre en español: Eficiencia esencial

Nombre en inglés: Essential efficiency

Descripción: La métrica trata de medir qué tan cerca está el diseño de la interfaz de usuario del caso de uso ideal para hacer una tarea concreta. Es decir, trata de una relación entre el número de pasos esenciales que necesitan para un usuario para conseguir un objetivo y el número de pasos realmente necesarios para conseguirlo

Fórmula:

$$EE = 100 * (A/B), \text{ Donde}$$

A: número de pasos esenciales,
B: Número de pasos establecidos

Atributo al que ayuda a calcular: Usabilidad

Referencias: Penichet [60]

Espectro: Gris

Otros: Aplica en las disciplinas de análisis

MÉTRICA 3

Nombre en español: Concordancia de tarea

Nombre en inglés: Task concordance

Descripción: Basado en los casos de uso y evalúa la eficiencia de trabajo y la sencillez. Es un indicador sobre la dificultad de las tareas más frecuentes. De esta manera, un sistema sería mejor que otros sistemas si las tareas más frecuentemente utilizadas son las más simples. Valores altos (Cerca a 100% o 100%) indican que las tareas de uso más frecuente son las más simples. Por otra parte valores muy bajos (-100%), indican que tareas más frecuentemente utilizadas son las tareas que necesitan el mayor número de pasos para completar.

Fórmula:

$$TC = 100 * (D/P), \text{ donde,}$$

D: número de pares de tareas clasificadas en el orden correcto promulgado por el número de menor longitud de pares fuera de orden.
P: número de pares de tareas posibles

Atributo al que ayuda a calcular: Usabilidad

Granularidad: programa

Rango: -100 a 100

Referencias: Penichet [60]

Espectro: Gris

Otros: Aplica en las disciplinas de análisis, pruebas y Mantenimiento

MÉTRICA 4

Nombre en español: Visibilidad de tarea

Nombre en inglés: Task visibility

Descripción: La métrica esta basada en el principio de visibilidad (las interfaces de usuario sólo deben mostrar lo que los usuarios necesitan para hacer una cosa en particular). Mide la relación entre la visibilidad de las características y la capacidad necesaria para terminar una tarea o un conjunto de tareas

Fórmula:

$$TV = 100 * \{ [1/Stotal] * [\sum \text{para todo } i \text{ de } (Vi)] \}, \text{ Donde,}$$

Stotal: Número de pasos para completar los casos de uso
Vi: función de la visibilidad (0 a 1) del paso i

Atributo al que ayuda a calcular: Usabilidad

Granularidad: Programa

Referencias: Penichet [60]

Espectro: Oscura

Otros: Aplica en las disciplinas de diseño, implementación, análisis y Mantenimiento

MÉTRICA 5

Nombre en español: Uniformidad de distribución

Nombre en inglés: Layout uniformity

Descripción: Debido a que en general los desarrolladores no tienen habilidad en diseño gráfico, entonces, esta medida métrica de la disposición de los diferentes objetos en la interfaz de usuario sin tener en cuenta la funcionalidad de estos elementos, lo que son o lo que hacen

Fórmula:

$$LU = 100 * [1 - \{ (Nh + Nw + Nt + NI + Nb + Nr) - M \} / (6 * Nc - M)], \text{ donde,}$$

Nc: Número de componentes visuales en la interface,
Nh, Nw, Nt, NI, Nb, Nr: alturas, anchuras, alineaciones... entre los componentes.
M: Este es un ajuste para hacer el valor de la uniformidad de distribución de 0 a 100.

Atributo al que ayuda a calcular: Usabilidad

Granularidad: Programa

Referencias: Penichet [60]

Espectro: Oscura

Otros: Las disciplinas en las que aplica son diseño, análisis y Mantenimiento

MÉTRICA 6

Nombre en español: Coherencia visual

Nombre en inglés: Visual coherence

Descripción: Basada en la distancia entre los componentes visuales relacionados, y los trabajos teniendo en cuenta que para los usuarios es más fácil encontrar lo que está buscando en una interfaz de usuario bien

estructurada. Esto sucede porque todos los artículos relacionados están cerca unos de otros y los artículos no relacionados son distantes.

Fórmula:

$VC=100* \{[\sum (G_k) \text{ para todo } k] / [\sum \text{para todo } k \text{ de } (N_k*(N_k-1)/2)]\}$, donde,

$G_k = [\sum (R_{i,j})]$, para todo $i, j (i \neq j)$,

N_k : número de componentes visuales en el grupo k .

$R_{i,j}$: relación semántica entre los componentes i y j en el conjunto k , $0 \leq R_{i,j} \leq 1$.

Atributo al que ayuda a calcular: Usabilidad

Granularidad: Programa

Referencias: Penichet [60]

Espectro: Oscura

Otros: Las disciplinas en las que Aplica son diseño, análisis y Mantenimiento

MÉTRICA 7

Nombre en español: Disponibilidad de construcción y pruebas de función

Nombre en inglés: Availability of built-in test function

Descripción: ¿Pueden usuario y mantenedor realizar fácilmente pruebas operacionales sin preparación de motor de pruebas adicionales? Para esto observar el comportamiento del usuario o mantenedor quien ensaya el sistema software después del Mantenimiento.

Fórmula:

$X=A/B$, donde,

A (conteo): Número de casos en donde el mantenedor puede usar adecuadamente la construcción en función de la prueba.

B (conteo): número de casos de oportunidades de prueba.

Atributo al que ayuda a calcular: Testeabilidad

Influencia sobre otros atributos:

Granularidad: programa

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 es mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

Otros: Aplica en las disciplinas de pruebas y Mantenimiento

MÉTRICA 8

Nombre en español: Eficiencia de re – prueba

Nombre en inglés: Re-test efficiency

Descripción: ¿Pueden usuario y mantenedor realizar fácilmente pruebas operacionales y determinar si el software esta o no listo para operar? Para esto observar el comportamiento del usuario o mantenedor quien ensaya el sistema software después del Mantenimiento.

Fórmula:

$X=\sum (T)/N$, donde,

T (tiempo): tiempo gastado en pruebas, para estar seguro de si las fallas reportadas fueron o no resueltas.

N (Conteo): Número de fallas resueltas.
Atributo al que ayuda a calcular: Testeabilidad
Granularidad:
Rango: $0 < X$, entre más pequeño mejor.
Referencias: ISO 9126 [15]:2
Espectro: Gris
Otros: Aplica en las disciplinas de pruebas y Mantenimiento. X
(Tiempo/Conteo)

MÉTRICA 9

Nombre en español: Prueba de reactivación
Nombre en inglés: Test restartability
Descripción: ¿Pueden usuario y mantenedor realizar fácilmente pruebas operacionales con puntos de chequeo después del Mantenimiento? Para esto observar el comportamiento del usuario o mantenedor quien ensaya el sistema software después del Mantenimiento.

Fórmula:

$$X = A / B, \text{ donde,}$$

A= Número de casos en los cuales el mantenedor puede hacer pausa y reiniciar la prueba de ejecución, en los puntos deseados paso por paso

B= Número de casos de pausa en el teste de ejecución.

Atributo al que ayuda a calcular: Testeabilidad

Influencia sobre otros atributos:

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 es mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

Otros: Puede aplicar en la disciplina de pruebas

MÉTRICA 10

Nombre en español: Completitud de construcción en prueba de función

Nombre en inglés: Completeness of built-in test function

Descripción: ¿Qué tan completa es la capacidad del test integrado?

Cuenta el número de funciones de test integradas implementadas como específicas y las compara al número de funciones de test integradas en los requerimientos

Fórmula:

$$X = A/B, \text{ Donde,}$$

A (conteo): Número de funciones de test integradas implementadas como confirmadas específicas en revisión.

B (conteo): número funciones de test integradas requeridas.

Atributo al que ayuda a calcular: Testeabilidad

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 es mejor.

Referencias: ISO 9126 [15]:3

Espectro: Gris

Otros: Aplica en disciplina de implementación

MÉTRICA 11

Nombre en español: Autonomía de capacidad de prueba

Nombre en inglés: Autonomy of testability

Descripción: ¿Qué tan independientemente puede el software ser probado? Cuenta el número de dependencias sobre otros sistemas para experimentación que han sido simulados con raíces y los compara con el número total de dependencias probadas en otros sistemas

Fórmula:

$$X=A/B,$$

A (conteo): Número de dependencias sobre otros sistemas para experimentación que ha sido simulada con raíces

B (conteo): número total de dependencias de otros sistemas

Atributo al que ayuda a calcular: Testeabilidad

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 es mejor.

Referencias: ISO 9126 [15]:3

Espectro: Gris

Otros: Aplica en disciplinas de diseño e implementación

MÉTRICA 12

Nombre en español: Observación del progreso de prueba

Nombre en inglés: Test progress observability

Descripción: ¿Qué tan completos son los resultados del test integrado, mostrados durante la prueba? Cuenta el número de puntos de chequeo implementados como fueron especificados y los compara con el número requerido de puntos de chequeo especificados en diseño

Fórmula:

$$X=A/B, \text{ donde,}$$

A (conteo): Número de puntos de chequeo implementados como fueron especificados, confirmados en revisión.

B (conteo): número puntos de chequeo diseñados.

Atributo al que ayuda a calcular: Testeabilidad

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 es mejor.

Referencias: ISO 9126 [15]:3

Espectro: Gris

Otros: Aplica en disciplina de pruebas

MÉTRICA 13

Nombre en español: InterCambiabilidad de datos (Basado en formato de datos)- externa

Nombre en inglés: Data exchangeability (Data format based)

Descripción: ¿Cuan correctamente tiene las funciones de interface intercambiadas para datos de transferencia especificados a ser

implementados? hacer prueba de cada formato de registro de salida de la interface descendiente del sistema de acuerdo a las como en especificaciones de los campos de los datos. Cuenta el número de formatos de datos que son aprobados para ser intercambiados con otro software o sistema durante las pruebas sobre datos intercambiables en comparación con el número total.

Fórmula:

$$X=A/B, \text{ donde,}$$

A (conteo): Número de formatos de datos que son aprobados para ser intercambiados satisfactoriamente con otro software o sistema durante las pruebas sobre datos intercambiables.

B (conteo): número total de formatos de datos a ser intercambiados.

Atributo al que ayuda a calcular: Interoperabilidad

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 es mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

Otros: Aplica en disciplina de pruebas

MÉTRICA 14

Nombre en español: InterCambiabilidad de datos (Basado en el intento de éxito del usuario)

Nombre en inglés: Data exchangeability (User's success attempt based)

Descripción: ¿Qué tan seguido falla el usuario final en intercambiar datos entre el software anotado y otro software? ¿Cuán seguido son satisfactorias las transferencias de datos entre el software anotado y otro software? Cuenta el número de casos para los que las funciones de interface son usadas y fallan

Fórmula:

$$\text{a) } X=1 - A/B \qquad \text{b) } Y=A/T, \text{ donde,}$$

A (Conteo): Número de casos en los cuales el usuario falla para intercambiar datos en otro software o sistemas.

B (Conteo): número de casos en los cuales el usuario intenta intercambiar datos.

T (Conteo): periodos de tiempo de operación.

Atributo al que ayuda a calcular: Interoperabilidad

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 es mejor. $0 \leq Y$, entre más cerca a cero mejor

Referencias: ISO 9126 [15]: 2

Espectro: Gris

Otros: Aplica en disciplina de pruebas

MÉTRICA 15

Nombre en español: InterCambiabilidad de datos (Basado en formato de datos)- interna

Nombre en inglés: Data exchangeability (Data format based)

Descripción: ¿Qué tan correctamente los formatos de dato de la interface estan implementado? Cuenta el número de formatos de dato de la interface

que han sido implementados correctamente como en las especificaciones y compara al número de formatos de datos a ser intercambiados como en las especificaciones

Fórmula:

$X=A/B$, donde,

A (conteo): Número de formatos de datos de interface que han sido implementados correctamente como en las especificaciones.

B (conteo): número de formatos de datos a ser intercambiados como en las especificaciones.

Atributo al que ayuda a calcular: Interoperabilidad

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 más correcto.

Referencias: ISO 9126 [15]: 3

Espectro: Gris

Otros: Aplica en disciplina de implementación

MÉTRICA 16

Nombre en español: Coherencia de interfaz (Protocolo)

Nombre en inglés: Interface consistency (protocol)

Descripción: ¿Qué tan correcto los protocolos de interface han sido implementados? Cuenta el número de protocolos de interface que fueron implementados correctamente como en las especificaciones y compara con el número de protocolos de interface a ser implementados como en las especificaciones.

Fórmula:

$X=A/B$, donde,

A (conteo): Número de protocolos de interface implementados con formato consistente como en la especificación confirmado en revisión.

B (conteo): número de protocolos de interface a ser implementados como en las especificaciones.

Atributo al que ayuda a calcular: Interoperabilidad

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 es más consistente

Referencias: ISO 9126 [15]: 3

Espectro: Gris

Otros: Aplica en la disciplina de implementación

MÉTRICA 17

Nombre en español: Número de hijos o descendientes (NDD)

Nombre en inglés: Number of children (NOC)

Descripción: La métrica es el número de subclases de una clase en la jerarquía. Es un indicador del nivel de Reutilización, la posibilidad de haber creado abstracciones erróneas, y es un indicador del nivel de pruebas requerido. Un mayor número de hijos requiere más pruebas de los métodos de esa clase y aumenta la dificultad para modificar la clase pues afecta a todos los hijos de dicha clase. Clases en un nivel más alto en la jerarquía deberían

tener más subclases que las clases en un nivel más bajo en la jerarquía. NOC puede ser también un indicador del uso inadecuado de la Herencia. Es un potencial indicador de la influencia que una clase puede tener sobre el diseño del sistema. Si el diseño depende mucho de la Reutilización a través de la Herencia, quizás sea mejor dividir la Funcionalidad en varias clases.

Fórmula:

NOC= Número de subclases inmediatamente subordinadas de una clase en la jerarquía de clases,

Atributo al que ayuda a calcular: Reusabilidad, Herencia

Influencia sobre otros atributos: Analizabilidad, Cambiabilidad

Granularidad: Clase

Rango: 0 hasta N, donde N es un entero positivo

Referencias: OLMEDILLA [53], Vazquez [61], Chidamber y Kemerer [67], Gonzalez [68], PRESSMAN [1], Pritchett [69], Basili [70], El-Wakil [71], Li [72], Lindroos [73], Andersson [74], Bar [75], Kan [24], Laird [76], Fenton [77], Garzas [57], Marin [28]

Espectro: Clara

Otros: Aplica en disciplina de diseño. Es la misma métrica de NOD Number Of Descendants (Métrica 135)

MÉTRICA 18

Nombre en español: Beneficio de Reutilización

Descripción: Mide el beneficio de reutilizar en un sistema (S)

Fórmula:

$Rb(S) = [Csin_reutilizacion - Ccon_reutilizacion] / Csin_reutilizacion$, donde,
Csin_reutilizacion: coste de desarrollar S sin Reutilización,
Ccon_reutilizacion: coste de desarrollar con Reutilización.

Atributo al que ayuda a calcular: Reusabilidad

Rango: $0 \leq Rb \leq 1$

Referencias: PRESSMAN [1]

Espectro: Oscura

MÉTRICA 19

Nombre en español: Aprovechamiento de la Reutilización

Fórmula:

$Raprovechado = OBJreutilizados / OBJconstruidos$, donde
OBJreutilizados: Número de objetos reutilizados en el sistema;
OBJconstruidos: Número de objetos construidos para el sistema, si aumenta

Atributo al que ayuda a calcular: Reusabilidad

Referencias: PRESSMAN [1]

Espectro: Oscura

Otros: Aprovechado también aumenta la Reutilización

MÉTRICA 20

Nombre en español: Acoplamiento entre objetos

Nombre en inglés: Coupling Between objects (CBO)

Descripción: Es el número de clases a las cuales una clase está acoplada, sin tener con ella relaciones de herencia (hijo o padre). Hay dependencia entre dos clases cuando una de ellas usa métodos o variables de la otra clase. Es consistente con las tradicionales definiciones de acoplamiento: "medida del grado de interdependencia entre módulos". Es un indicador del esfuerzo necesario para Mantenimiento y pruebas (considera que un objeto está acoplado a otro si una de sus acciones está en el otro).

Fórmula:

CBO=Número de acoplamientos de una clase

Atributo al que ayuda a calcular: Reusabilidad, acoplamiento

Influencia sobre otros atributos: Complejidad, modularidad, Encapsulamiento, Analizabilidad, Cambiabilidad, Estabilidad

Granularidad: Clase, programa, sistema

Referencias: Manso [85], VAZQUEZ [61], Chidamber y Kemerer [67], Garzas [57], OLMEDILLA [53], Manso [85], Basili [70], El-Wakil [71], Ojha [78], Lindroos [73], Andersson [74], Bar [75], LAIRD [76], Fenton [77], Marin [28]

Espectro: Claro

Otros: Aplica en la disciplina de diseño. Es sugerido como un indicador de esfuerzo para Mantenimiento y pruebas. Cuanto más independiente es un objeto es más fácil de reutilizar. La métrica también puede verse como el número de mensajes enviados entre dos clases. Tiene relación con la métrica DAC (Métrica 28)

MÉTRICA 21-A

Nombre en español: Carencia de Cohesión en los métodos (CMM)

Nombre en inglés: Lack of cohesion in methods (LCOM)

Descripción: La métrica es el número de métodos locales que no accedan a atributos comunes. Indica la cantidad de abstracción hecha en una clase. Si no hay atributos comunes el grado de similitud es cero. Es preferible una alta cohesión en los métodos dentro de una clase, ya que esta no puede dividirse fomentando el encapsulamiento, una baja cohesión tiende a incrementar la complejidad.

Fórmula:

Considere una clase C1 con n métodos M1, M2,..., Mn.

Sea {Ij}= conjunto de variables instancia usadas por el método Mi.

Hay n conjuntos {I1},..., {In}.

Sea P= {(Ii, Ij) | Ii (intersección) Ij=0} y

Q = {(Ii, Ij) | Ii (intersección) Ij≠0}.

Si todo n conjuntos {I1},..., {In} son 0 entonces P=0,

LCOM=|P| - |Q|, si |P|>|Q|;

De otra manera LCOM=0.

Atributo al que ayuda a calcular: Reusabilidad, Cohesión, Complejidad

Influencia sobre otros atributos: Encapsulamiento, Analizabilidad, Cambiabilidad, Estabilidad

Granularidad: Clase

Rango: 0 hasta N, N entero positivo

Referencias: VAZQUEZ [61], Chidamber y Kemerer [67], GONZALEZ [68], PRESSMAN [1], OLMEDILLA [53], Lindell [86], PRITCHETT [69] Basili [70], EI-Wakil [71], Li [72], Ojha [78], Lindroos [73], ANDERSSON [74], Bar [75], Kan [24], LAIRD [76], Fenton [77], Garzas [57], Marin [28], Manso [85]

Espectro: Gris

Otros: Llamado por Pritchett [69] como "class cohesion" (Métrica 442). Otra definición LCOM= número de conjuntos disjuntos de los métodos locales; no dos intersecciones; cualquiera dos métodos en el mismo conjunto compartido al menos un variable de instancia local. Un inconveniente de esta métrica es que no posee una guía clara para su uso.

MÉTRICA 21-B

Nombre en español: Carencia de cohesión en los métodos** (CMM**)

Nombre en inglés: Lack of cohesion in methods** (LCOM**)

Descripción: Misma definición de Métrica 21-A. Para el caso de cohesión perfecta, LCOM** será igual a 0, y en el caso contrario (Clase totalmente incoherente) el valor de LCOM** será 1.

Fórmula:

$$LCOM^{**} = \{media [p (f)]\} / (1-m), \text{ donde,}$$

m: es el número de métodos definidos en la clase,
Sea F el conjunto de atributos definidos por la clase.

Entonces

p (f): es el número de métodos que accesan al atributo f, donde,
f: es miembro de F

Atributo al que ayuda a calcular: Reusabilidad, Cohesión

Granularidad: Clase

Rango: 0 hasta 1

Referencias: Andersson [74]

Espectro: Gris

MÉTRICA 22

Nombre en español: Número promedio de antecesores

Nombre en inglés: Average number of ancestors (ANA)

Descripción: El valor de esta métrica es el número promedio de clases de las que una clase hereda información. Es calculada determinando el número de clases en todos los caminos desde la clase(s) raíz a todas las clases en una estructura de herencia.

Fórmula:

$$ANA (Ci) = \frac{\{\sum \text{ para } j=1 \text{ hasta } TC \text{ de } [Hereda (Ci, Cj)]\}}{TC}$$

Donde:

Hereda (Ci, Cj) = 1 si y solo si Cj heredas a Ci, 0 en caso contrario

TC: es el número total de clases,

Ci: Clase en análisis

Cj. Clases que pueden heredar a Ci

Atributo al que ayuda a calcular: Abstracto (Abstracción)

Granularidad: Clase

Referencias: Bansiya [31], JETTER [3], OLMEDILLA [53], Garzas [57]

Espectro: Gris

Otros: Aplica en la disciplina de diseño. Según JETTER [3] puede ser remplazada por la métrica "Abstractness"

MÉTRICA 23

Nombre en español: Abstracción

Nombre en inglés: Abstractness

Descripción: El grado de uso de conceptos de Abstracción. Relaciona el número de clases abstractas al total de clases en el paquete. A= 0 significa que el paquete no contiene tipos abstractos. A=1, significa que el paquete no contiene más que los tipos abstractos

Fórmula:

$$A = Na / Nc, \text{ donde,}$$

Na: Número de tipos abstractos (clases) en el paquete.

Nc: número total de clases en el paquete.

Atributo al que ayuda a calcular: Abstracto

Influencia sobre otros atributos: métrica de "Instability"

Granularidad: Paquete

Rango: desde 0 hasta 1

Referencias: JETTER [3], Andersson [74]

Espectro: Gris

Otros: Aplica en disciplina de diseño

MÉTRICA 24

Nombre en español: Lista de chequeo de Auditoria de acceso

Nombre en inglés: Access audit checklist

Descripción: La métrica es el puntaje promedio de los siguientes elementos:

1. Provisiones para registro y reporte de acceso: una declaración de los requerimientos para este tipo de capacidad debe existir en la especificación de requerimientos. Esto es considerado en la especificación del diseño, y codifica durante la implementación. esta es una métrica binaria aplicada a todas las tres fases identificadas si estos pasos son tomados. ejemplos de provisiones que pueden ser considerados estarían registrados en los vínculos terminales, accesos de archivos de datos, y trabajos activados por la identificación de usuario y tiempo

2. Provisiones para indicación inmediata de violación de acceso: además de (1), las capacidades de auditoría de acceso requeridas pueden incluir no solo acceso de registro sino identificación inmediata de acceso no autorizado, intencional o no. esta medida traza el requerimiento , diseño e implementación de provisiones para esta capacidad

Fórmula: A nivel de sistema:

$$X = A / B, \text{ donde,}$$

A: Puntajes totales de los elementos aplicables,

B: Número de elementos aplicables.

Atributo al que ayuda a calcular: Auditoria de acceso

Granularidad: Sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica a disciplinas de análisis de requerimientos, diseño e implementación a nivel de sistema

MÉTRICA 25

Nombre en español: Clase de acoplamiento directo

Nombre en inglés: Direct class coupling (DCC)

Descripción: Esta métrica es una cuenta del diferente número de clases que tiene una clase directamente relacionada. La métrica incluye clases que son directamente relacionadas por declaraciones de atributos y paso de mensajes (parámetros) en métodos

Fórmula:

$$DCC (C_i) = \text{Número de clases directamente relacionadas a la clase } C_i$$

Atributo al que ayuda a calcular: Acoplamiento

Granularidad: Clase

Referencias: Bansiya [31], JETTER [3], OLMEDILLA [53], Garzas [57]

Espectro: Gris

Otros: Esta métrica puede ser remplazada por la métrica "Instability" (métrica 452) según JETTER [3]

MÉTRICA 26

Nombre en español: Factor de acoplamiento

Nombre en inglés: Coupling Factor - COF

Descripción: Proporción entre el número real de acoplamientos no imputables a herencia y el máximo número posible de acoplamientos en el sistema. Es decir indica la comunicación entre clases.

Fórmula:

$$COF = \frac{\sum_{i=1}^{TC} \text{de } (\sum_{j=1}^{TC} \text{de } (es_cliente (C_i, C_j))))}{(TC^2) - TC}$$

Donde

(TC²)-TC: es el máximo número de acoplamientos en un sistema con TC clases,

es_cliente (Ci, Cj): es 0 o 1, 1 (si y solo si Cc Cs Cc Cs), 0 caso contrario

La relación cliente-servidor (Cc->Cs) significa que Cc (la clase cliente) contiene al menos una referencia no basada en la herencia a una característica (método o atributo) de la clase Cs (clase proveedora). El numerador representa el número real de acoplamientos no imputables a la herencia. El denominador representa el máximo número posible de acoplamientos en un sistema con TC clases. Las relaciones cliente servidor Pueden tener distintas formas:

Paso de mensajes regular.

Paso de mensajes "forzado".

Iniciación y destrucción de objetos.

Asociaciones semánticas entre clases con una cierta relación (p.e. 1: 1, 1: n o n: m).

Atributo al que ayuda a calcular: Acoplamiento

Influencia sobre otros atributos: Medida indirecta de: Complejidad, Encapsulamiento, Reutilización, Extensibilidad, Mantenimiento, Analizabilidad, Cambiabilidad, Estabilidad

Granularidad: Clase, programa, sistema

Referencias: VAZQUEZ [61], Abreu y Melo [79], PRESSMAN [1], Champeaux [80], Marin [28]

Espectro: Claro

Otros: Aplica en disciplina de diseño.

MÉTRICA 27

Nombre en español: Acoplamiento de otros módulos

Nombre en inglés: Coupling to other modules

Descripción: Esta métrica cuenta el número de módulos, junto a otros módulos y los relaciona con el número total de módulos. Afecta la reutilización de los módulos

Fórmula:

$$X = A / B, \text{ donde}$$

A: número de módulos importados desde otros módulos,

B: número total de módulos

Atributo al que ayuda a calcular: Acoplamiento

Influencia sobre otros atributos: Reusabilidad

Referencias: Zeiss [56]

Espectro: Gris

MÉTRICA 28

Nombre en español: Acoplamiento de Abstracción de datos

Nombre en inglés: Data abstraction coupling - DAC

Descripción: Es calculada como el número de las declaraciones de la "clase X" en una clase. Este indicador puede ser más o menos a partir de la información disponible en el modelo de objetos. Esto puede volver a ser aplicado sobre la base de la "compuesto por" las relaciones entre las clases. El uso del modelo de objetos es otra vez una metodología específica

Fórmula:

DAC= El número de atributos en una clase que tienen como tipo otra clase (Atributos de otra clase, es decir atributos heredados).O. Número de ADTs definidos en una clase,
ADT: Tipo abstracto de dato

Atributo al que ayuda a calcular: Acoplamiento

Influencia sobre otros atributos: Abstracción, Analizabilidad, Cambiabilidad, Estabilidad

Granularidad: clase

Referencias: OLMEDILLA [53], Lindell [86], Manso [85], Li [72], OJHA [78], Bar [75], Garzas [57], Marin [28]

Espectro: Gris

Otros: Aplica en la disciplina de diseño, relacionada con las métricas RFC (Métrica 266), CBO (Métrica 20), NIV (Métrica 395)

MÉTRICA 29

Nombre en español: Acoplamiento de paso de masajes

Nombre en inglés: Message passing coupling - MPC

Descripción: Es una cuenta de el número total de llamadas (funciones y procedimientos) hechas a unidades externas. Diferentes llamadas hechas de una misma rutina son contadas separadamente. Esta medida cuantifica la extensión en que la clase es acoplada a otras clases. Un alto nivel de acoplamiento es generalmente considerado problemático en esos cambios en una clase pueden afectar todas las clases a que esta acoplada.

Fórmula:

MPC= número de métodos invocados en una clase

Atributo al que ayuda a calcular: Acoplamiento

Influencia sobre otros atributos: Analizabilidad

Granularidad: Clase

Referencias: OLMEDILLA [53], Lindell [86], PRITCHETT [69], Li [72], Bar [75], Garzas [57], Marin [28]

Espectro: Gris

Otros: Puede aplicar en disciplina de diseño

MÉTRICA 30

Nombre en español: Número de clases llamadas

Nombre en inglés: Number of Called Classes - FANOUT

Descripción: Esta métrica es calculada como la suma de las métricas FANOUT Lorenz & Kidd [87] (es decir, las clases desde las que las operaciones llaman métodos) para todas las operaciones definidas por el usuario. Este indicador proporciona información escasa acerca de cómo llamadas a operaciones dispersas están en clases.

Atributo al que ayuda a calcular: Acoplamiento

Granularidad: Clase

Referencias: Lanza [88]

Espectro: Oscura

Otros: Es un indicador de que tanto acoplamiento involucran muchas clases.

MÉTRICA 31

Nombre en español: Número de llamadas de operación

Nombre en inglés: Number of Operation Calls -CALLS

Descripción: Esta métrica cuenta el número total de llamadas a operaciones distintas (invocaciones) en el proyecto, sumando el número de operaciones llamadas por todas las operaciones definidas por el usuario. Si una operación de foo () es llamada tres veces por un método de f1 () cuenta una sola vez. Si es llamado por los métodos de f1 (), f2 () y F3 (), cuenta tres llamados para esta métrica

Atributo al que ayuda a calcular: Acoplamiento

Referencias: Lanza [88]

Espectro: Oscuro

Otros: Denota el nivel de acoplamiento entre operaciones, valores altos indican un excesivo acoplamiento entre operaciones

MÉTRICA 32

Nombre en español: Adaptabilidad de estructuras de datos-externo

Nombre en inglés: Adaptability of data structures

Descripción: ¿Puede usuario o mantenedor adaptar fácilmente el software a los conjuntos de datos en un nuevo entorno? observar el comportamiento de usuario o mantenedor cuando esté tratando de adaptar el software a un entorno operacional.

Fórmula:

$$X = A / B, \text{ donde}$$

A (conteo): Número de datos los cuales son operables, aunque no son observados debido a operaciones incompletas causadas por limitaciones de adaptación.

B (conteo): número de datos los cuales espera sean operables en el entorno en el cual el software es adaptado.

Atributo al que ayuda a calcular: Adaptabilidad

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 es mejor.

Referencias: ISO 9126 [15]: 2

Espectro: Gris

Otros: Puede usarse en disciplina de pruebas

MÉTRICA 33

Nombre en español: Adaptabilidad de entorno hardware - externo

Nombre en inglés: Hardware environmental adaptability (adaptability to hardware devices and network facilities)

Descripción: ¿Puede usuario o mantenedor adaptar fácilmente el software al entorno? ¿El sistema software es lo suficiente capaz de adaptarse a entornos operacionales? observar el comportamiento de usuario o mantenedor cuando está tratando de adaptar el software a un entorno operacional.

Fórmula:

$$X = 1 - A/B,$$

A (conteo): Número de funciones operativas de tareas que no fueron completadas o no resultaron suficientes para satisfacer niveles adecuados durante pruebas operativas combinadas con entornos hardware.

B (conteo): número total de funciones que son probadas

Atributo al que ayuda a calcular: Adaptabilidad

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 es mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

Otros: Puede usarse en disciplina de pruebas y mantenimiento

MÉTRICA 34

Nombre en español: Adaptabilidad de entorno organizacional -externo

Nombre en inglés: Organisational environment adaptability

Descripción: ¿Puede usuario o mantenedor adaptar fácilmente el software al entorno? ¿El sistema software es lo suficiente capaz de adaptarse a entornos operacionales? observar el comportamiento de usuario o mantenedor cuando está tratando de adaptar el software a un entorno operacional.

Fórmula:

$$X = 1 - A/B,$$

A (conteo): Número de funciones operativas de tareas que no fueron completadas o no resultaron suficientes para satisfacer niveles adecuados durante pruebas operativas con los entornos de negocio de usuario.

B (conteo): número total de funciones que son probadas.

Atributo al que ayuda a calcular: Adaptabilidad

Rango: X ($0 \leq X \leq 1$): entre grande es mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

Otros: Puede usarse en disciplina de pruebas

MÉTRICA 35

Nombre en español: Amigabilidad del puerto de usuario

Nombre en inglés: Porting user friendliness

Descripción: ¿Puede usuario o mantenedor adaptar fácilmente el software al entorno? observar el comportamiento de usuario o mantenedor cuando está tratando de adaptar el software a un entorno operacional.

Fórmula:

T (Tiempo) = Σ de tiempo operativo de usuario gastado para completar la adaptación del software para entornos de usuario, cuando el usuario intenta instalar o cambiar configuración.

T Implica "esfuerzo de usuario requerido para adaptar el entorno de usuario". Puede usarse Persona/hora en vez de tiempo.

Atributo al que ayuda a calcular: Adaptabilidad

Rango: $0 < T$, Entre más pequeño mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

Otros: Puede usarse en disciplina de pruebas y mantenimiento

MÉTRICA 36

Nombre en español: Adaptabilidad de entorno de sistema software –externo

Nombre en inglés: System software environmental adaptability (adaptability to OS, network software and cooperated application software)

Descripción: ¿Puede usuario o mantenedor adaptar fácilmente el software al entorno? ¿El sistema software es lo suficiente capaz de adaptarse a entornos operacionales? observar el comportamiento de usuario o mantenedor cuando está tratando de adaptar el software a un entorno operacional.

Fórmula:

$$X = 1 - A/B,$$

A (conteo): Número de funciones operativas de tareas que no fueron completadas o no resultaron suficientes para alcanzar niveles adecuados durante pruebas operativas combinadas con software de sistema operativo o aplicaciones de software concurrente.

B (conteo): número total de funciones probadas.

Atributo al que ayuda a calcular: Adaptabilidad

Rango: X ($0 \leq X \leq 1$): entre grande es mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

Otros: puede usarse en disciplina de pruebas y mantenimiento

MÉTRICA 37

Nombre en español: Adaptabilidad de estructuras de datos-interno

Nombre en inglés: Adaptability of data structures

Descripción: Cuan adaptable es el producto a cambios de estructura de datos Cuenta el número de estructuras de datos, las cuales son estructuras operables y no tienen limitación después de la adaptación y las compara con el número total de estructuras de datos que requieren capacidad de adaptación

Fórmula:

$$X=A/B,$$

A (conteo): Número de estructuras de datos las cuales son operables y no tienen limitación después de la adaptación, confirmación en revisión.

B (conteo): número total de estructuras que requieren capacidad de adaptación.

Atributo al que ayuda a calcular: Adaptabilidad

Rango: X ($0 \leq X \leq 1$): más cerca a 1 es mejor.

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 38

Nombre en español: Adaptabilidad de entorno hardware-interno

Nombre en inglés: Hardware environmental adaptability

Descripción: Cuan adaptable es el producto a cambios ambientales relacionados con H/W. Cuenta el número de funciones implementadas las cuales son capaces de mejorar resultados requeridos en ambientes H/W múltiples concretos, como fueron especificados y los compara al número de funciones con requerimientos de capacidad de adaptación de ambientes H/W

Fórmula:

$$X=A/B,$$

A (conteo): Número de funciones implementadas las cuales son capaces de mejorar resultados requeridos en ambientes H/W múltiples concretos como especificados, confirmación en revisión.

B (conteo): número total de funciones con requerimiento de capacidad de adaptación en ambientes H/W.

Atributo al que ayuda a calcular: Adaptabilidad

Rango: X ($0 \leq X \leq 1$): más cerca a 1 es mejor.

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 39

Nombre en español: Adaptabilidad de entorno organizacional-interno

Nombre en inglés: Organisational environment adaptability (organisation adaptability to infrastructure of organization)

Descripción: ¿Qué tan adaptable es el producto a cambios organizacionales? Cuenta el número de funciones implementadas las cuales son capaces de mejorar resultados requeridos en organizaciones múltiples concretas y ambientes de negocio específicos y los compara al número de funciones con los requerimientos de capacidad de adaptación de ambientes organizacionales

Fórmula:

$$X=A/B,$$

A (conteo): Número de funciones implementadas las cuales son capaces de mejorar resultados requeridos en ambientes organizacionales y de negocios concretos como especificados, confirmación en revisión.

B (conteo): número total de funciones con requerimientos de capacidad de adaptación en ambientes organizacionales.

Atributo al que ayuda a calcular: Adaptabilidad

Rango: X ($0 \leq X \leq 1$): más cerca a 1 es mejor.

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 40

Nombre en español: Amigabilidad de usuario de puerto-interno

Nombre en inglés: Porting user friendliness

Descripción: Cuan poco esfuerzo para desempeñar operaciones de porting en el producto Cuenta el número de funciones implementadas las cuales son capaces de soportar el fácil-de-adaptar por el usuario como especificado y lo compara con el número de funciones con requerimientos de capacidad fácil-de-adaptar

Fórmula:

$$X=A/B,$$

A (conteo): Número de funciones de soporte Fácil-de- adaptar por usuario como especificado, confirmado en revisión.

B (conteo): número total de funciones con requerimiento de capacidad de Fácil-de- adaptar

Atributo al que ayuda a calcular: Adaptabilidad

Rango: ($0 \leq X \leq 1$): entre más cerca a 1 es más amigable.

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 41

Nombre en español: Adaptabilidad de entorno de sistema software –interno

Nombre en inglés: System software environmental adaptability

Descripción: ¿Qué tan adaptable es el producto a cambios ambientales relacionados al sistema software? Cuenta el número de funciones implementadas las cuales son capaces de mejorar resultados requeridos en ambientes de sistemas software concreto como especificado y lo compara al número de funciones con los requerimientos de capacidad de adaptación de ambientes de sistema software

Fórmula:

$$X=A/B,$$

A (conteo): Número de funciones implementadas las cuales son capaces de mejorar resultados requeridos en ambientes de sistemas software múltiple concreto como especificados, confirmación en revisión.

B (conteo): número total de funciones con requerimiento de capacidad de adaptación en ambientes de sistemas software. B= número total de funciones con requerimientos de capacidad de adaptación de ambientes de sistemas software

Atributo al que ayuda a calcular: Adaptabilidad

Rango: X ($0 \leq X \leq 1$): más cerca a 1 es mejor.

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 42

Nombre en español: Cohesión entre los métodos en la clase

Nombre en inglés: Cohesion among methods in class (CAM)

Descripción: Esta métrica calcula la relevancia entre métodos de una clase basada en lista de parámetros de los métodos. La métrica es calculada usando la sumatoria de la intersección de parámetros de un método con la máxima independencia del conjunto de todos los tipos de parámetro en una clase

Atributo al que ayuda a calcular: Cohesión

Granularidad: método

Referencias: Bansiya [31], JETTER [3], OLMEDILLA [53], Garzas [57]

Espectro: Gris

Otros: Puede ser remplazada por la métrica "Lack of Cohesion of Methods" (Métrica 21-A)

MÉTRICA 43

Nombre en español: Medida de agregación

Nombre en inglés: Measure of aggregation (MOA)

Descripción: Esta métrica mide la extensión de toda parte de la relación, realizado por usar atributos. La métrica es una cuenta del número de declaraciones de datos cuyos tipos son definidos por las clases usuario

Fórmula:

MOA: Número de declaraciones de datos definidos por el usuario

Atributo al que ayuda a calcular: Composición

Granularidad: Clase

Referencias: Bansiya [31], JETTER [3], OLMEDILLA [53], Garzas [57]

Espectro: Claro

Otros: Puede aplicarse en disciplina de diseño. Según JETTER [3] puede ser remplazada por la métrica "Number of Attributes" (Métrica 44)

MÉTRICA 44

Nombre en español: Número de atributos

Nombre en inglés: Number of Attributes

Descripción: Es la cuenta del número total de atributos; tanto heredados como localmente definidos, para una clase. Esta medida es importante porque un

gran número de atributos puede indicar un pobre diseño (abstracción impropia), o que la clase es complicada y entonces más propensa a fallas.

Fórmula:

NA: Número total de atributos de la clase.

Atributo al que ayuda a calcular: Composición

Influencia sobre otros atributos: tamaño

Granularidad: Clase

Referencias: JETTER [3], Pritchett [69]

Espectro: Claro

Otros: O también llamada "total attributes", puede ser la misma que métrica 391

MÉTRICA 45

Nombre en español: Completitud 1

Nombre en inglés: Completeness 1

Descripción: ¿El programa requiere un equipo para inicialización del núcleo de almacenamiento previo a su utilización?

Atributo al que ayuda a calcular: Completitud

Granularidad: Programa

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 46

Nombre en español: Completitud 2

Nombre en inglés: Completeness 2

Descripción: ¿El programa contiene una facilidad para posicionamiento propio de dispositivos de entrada/salida previo a su utilización?

Atributo al que ayuda a calcular: Completitud

Granularidad: Programa

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 47

Nombre en español: Completitud 3

Nombre en inglés: Completeness 3

Descripción: ¿El programa contiene (1) código que específicamente predice frente a operaciones potencialmente indefinidas (e.g. división entre cero o raíz cuadrada de un número negativo) o (2) comentarios adecuados para definir las condiciones bajo las cuales las operaciones no definidas son posibles?

Atributo al que ayuda a calcular: Completitud

Granularidad: Programa

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 48

Nombre en español: Completitud 4

Nombre en inglés: Completeness 4

Descripción: ¿El programa contiene todos los programas referenciados no disponibles en la librería de sistema oficial?

Atributo al que ayuda a calcular: Completitud

Granularidad: Programa

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 49

Nombre en español: Completitud 5

Nombre en inglés: Completeness 5

Descripción: ¿El programa contiene servicios de control de reinicio?

Atributo al que ayuda a calcular: Completitud

Granularidad: Programa

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 50

Nombre en español: Completitud 6

Nombre en inglés: Completeness 6

Descripción: ¿El programa tiene la capacidad para asignar valores por defecto para parámetros no especificados?

Atributo al que ayuda a calcular: Completitud

Granularidad: Programa

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 51

Nombre en español: Completitud 7

Nombre en inglés: Completeness 7

Descripción: ¿Es chequeado el dato de entrada por rangos de errores?

Atributo al que ayuda a calcular: Completitud

Granularidad: Programa

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 52

Nombre en español: Completitud 8

Nombre en inglés: Completeness 8

Descripción: ¿Son probados los ciclos y rangos de parámetros de índices de transferencia múltiple antes de usarse?

Atributo al que ayuda a calcular: Completitud

Granularidad: Programa
Referencias: Boehm [17]
Espectro: Oscuro

MÉTRICA 53

Nombre en español: Completitud 9
Nombre en inglés: Completeness 9
Descripción: ¿Son probados los rangos de subíndices antes de usarse?
Atributo al que ayuda a calcular: Completitud
Granularidad: Programa
Referencias: Boehm [17]
Espectro: Oscuro

MÉTRICA 54

Nombre en español: Completitud 10
Nombre en inglés: Completeness 10
Descripción: ¿Son usadas todas las entradas al programa dentro del programa o su presencia explicada por un comentario?
Atributo al que ayuda a calcular: Completitud
Granularidad: Programa
Referencias: Boehm [17]
Espectro: Oscuro

MÉTRICA 55

Nombre en español: Completitud 11
Nombre en inglés: Completeness 11
Descripción: ¿Están “prohibidos” los nombres de variables impedidos (e.g. SQRT, ATAN, etc.)?
Atributo al que ayuda a calcular: Completitud
Granularidad: Programa
Referencias: Boehm [17]
Espectro: Oscuro

MÉTRICA 56

Nombre en español: Completitud 12
Nombre en inglés: Completeness 12
Descripción: ¿Hay subprogramas no “ociosos” referenciados?
Atributo al que ayuda a calcular: Completitud
Granularidad: Programa
Referencias: Boehm [17]
Espectro: Oscuro

MÉTRICA 57

Nombre en español: Completitud 13

Nombre en inglés: Completeness 13

Descripción: ¿El programa depende sobre sistemas o rutinas de librerías únicas para una instalación en particular?

Atributo al que ayuda a calcular: Completitud

Granularidad: Programa

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 58

Nombre en español: Completitud 14

Nombre en inglés: Completeness 14

Descripción: ¿El programa permite recuperación en caso de errores no fatales?

Atributo al que ayuda a calcular: Completitud

Granularidad: Programa

Referencias: Boehm [17]

Espectro: Oscuro

Otros: Esta métrica podría ser relacionada con el atributo recuperabilidad.

MÉTRICA 59

Nombre en español: Completitud 15

Nombre en inglés: Completeness 15

Descripción: ¿Está el código libre de errores obvios?

Atributo al que ayuda a calcular: Completitud

Granularidad: Programa

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 60

Nombre en español: Requisitos de completitud (Funcionales)

Nombre en inglés: requirements completeness

Descripción: De los requisitos funcionales

Fórmula:

$$Q2 = un / (ni * ns),$$

un: número de requisitos únicos de función,

ni: es el número de entradas,

ns: número de estados especificados.

Atributo al que ayuda a calcular: Completitud

Referencias: PRESSMAN [1]

Espectro: Gris

Otros: Esta métrica puede usarse en la disciplina de análisis

MÉTRICA 61

Nombre en español: Requisitos de completitud (No Funcionales)

Nombre en inglés: Requirements completeness

Descripción: De los requisitos no funcionales

Fórmula:

$$Q3 = nc / (nc * nnv),$$

nc: número de requisitos que han sido validados como correctos,

nnv: número de requisitos que no están validados todavía

Atributo al que ayuda a calcular: Completitud

Referencias: PRESSMAN [1]

Espectro: Gris

Otros: Esta métrica puede usarse en la disciplina de análisis

MÉTRICA 62

Nombre en español: Lista de chequeo de completitud

Nombre en inglés: Completeness Checklist

Descripción: Esta métrica es la suma de los puntajes de los siguientes elementos aplicables divididos por el número de elementos aplicables.

1. Referencias sin-ambigüedad (Entrada, función, salida): Referencias únicas para datos o funciones evitando ambigüedad tal como una función que tiene un nombre para un modulo y tiene otro nombre para otro modulo.
2. Todas las referencias de daos definidas: Cada elemento de dato debe tener un origen específico. Al nivel de requerimientos solo los mayores elementos de datos globales y algunos elementos de datos locales puede estar disponibles para ser revisados,
3. Todas las funciones definidas usadas: Una función la cual está definida pero no es usada durante una fase que no sea funcional o una referencia a la misma que ha sido omitida.
4. Todas las funciones referenciadas definidas: Un sistema no está completo en alguna fase si las funciones ficticias están presentes o si las funciones han sido referenciadas pero no definidas
5. Todas las condiciones y procesamientos definidos por cada punto de decisión: Cada punto de decisión tiene todas sus condiciones y partes de procesamiento alternativas definidas para cada fase del desarrollo del software, el nivel de detalle al que las condiciones y alternativas de procesamiento son descritas, pueden variar pero el elemento importante es el que todas las alternativas son explicadas.
6. Todas las llamadas definidas y referenciadas de acuerdo a parámetros de secuencia: Para cada interacción entre dos módulos, el complemento total de los parámetros definidos para la interfaz que va a

usarse. Una llamada particular a un modulo no pasara, por ejemplo, solo 5 de los 6 paramentaros definidos para aquel modulo.

7. Todos los reportes de problemas resueltos: En cada fase en el desarrollo, los reportes de problemas son generados. cada uno de ellos está cerrado, indica una resolución para asegurar un producto completo.
8. Diseño de acuerdo con requerimientos: Actualización continua de la documentación de requerimientos y la documentación de diseño es requerida para la versión actual del código fuente(elemento 9), la versión actual de la documentación de diseño, y la versión actual de la documentación de requerimientos acordados
9. Código de acuerdo con diseño: Ver elemento anterior (8)

Fórmula: A nivel de sistema:

$$X=A/B,$$

A = $\sum a_i$, (Para i de 1 hasta 9),

A_i (puntaje)= Puntuación del elemento i.

B =9, número total de elementos aplicables.

Atributo al que ayuda a calcular: Completitud

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica en disciplinas de análisis, diseño, implementación

MÉTRICA 63

Nombre en español: Completitud de descripción- Externo

Nombre en inglés: Completeness of description

Descripción: ¿Cuál es la proporción de funciones (o tipos de funciones) que son entendidas después de leer la descripción del producto? realizar una prueba de conducta de usuarios y entrevista al usuario con cuestionarios u observación de comportamiento. Cuenta el número de funciones que son adecuadamente entendidas y compara con el número de funciones del producto

Fórmula:

$$X=A/B,$$

A (conteo): Número de funciones (o tipos de funciones) entendidas.

B (conteo): número total de funciones (o tipos de funciones).

Atributo al que ayuda a calcular: Comprensibilidad

Rango: X (0<=X<=1): entre más cerca a 1 mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 64

Nombre en español: Accesibilidad de demostración

Nombre en inglés: Demonstration accessibility

Descripción: ¿Cuál es la proporción de demostraciones / tutoriales que puede acceder el usuario? realizar una prueba de de conducta de usuarios y observación de comportamiento de usuario. Cuenta el número de funciones que son adecuadamente demostrables y compara con el número de funciones que requieren capacidad de demostración

Fórmula:

$$X=A/B,$$

A (conteo): Número de demostraciones / tutoriales que el usuario accede satisfactoriamente.

B (conteo): número de demostraciones / tutoriales disponibles.

Atributo al que ayuda a calcular: Comprensibilidad

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 65

Nombre en español: Accesibilidad de demostración en uso

Nombre en inglés: Demonstration accessibility in use

Descripción: ¿Cuál es la proporción de demostraciones / tutoriales que puede acceder el usuario cuando realmente necesita hacerlo durante una operación? realizar observación de comportamiento de usuario, quien está tratando de ver la demostración / tutorial. La observación puede emplear monitoreo enfocado a acción cognitiva humana con cámara de video.

Fórmula:

$$X=A/B,$$

A (conteo): Número de casos en que el usuario ve satisfactoriamente la demostración cuando intenta verla.

B (conteo): número de casos en que el usuario intenta ver la demostración durante un periodo de observación.

Atributo al que ayuda a calcular: Comprensibilidad

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 mejor.

Referencias: ISO 9126 [15]: 2

Espectro: Gris

MÉTRICA 66

Nombre en español: Efectividad de demostración

Nombre en inglés: Demonstration effectiveness

Descripción: ¿Cuál es la proporción de funciones que puede acceder que el usuario pueda operar satisfactoriamente después de una demostración o tutorial? realizar observación de comportamiento de usuario, quien está tratando de ver la demostración / tutorial. La observación puede emplear monitoreo enfocado a acción cognitiva humana con cámara de video.

Fórmula:

$$X=A/B,$$

A (conteo): Número de funciones (o tipos de funciones) entendidas.

B (conteo): número total de funciones (o tipos de funciones).
Atributo al que ayuda a calcular: Comprensibilidad
Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 mejor.
Referencias: ISO 9126 [15]:2
Espectro: Gris

MÉTRICA 67

Nombre en español: Funciones evidentes- Externo
Nombre en inglés: Evident functions
Descripción: ¿Cuál es la proporción de funciones (o tipos de funciones) que pueden ser identificadas por el usuario basado en condiciones iniciales?. Realizar pruebas de conducta de usuario y entrevista de usuario con cuestionarios y observación de comportamiento. Cuenta el número de funciones que son evidentes para el usuario y compara con el número total de funciones.

Fórmula:

$$X=A/B,$$

A (conteo): Número de funciones (o tipo de funciones) identificadas por el usuario.

B (conteo): número total de funciones (o tipo de funciones).

Atributo al que ayuda a calcular: Comprensibilidad

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 68

Nombre en español: Comprensibilidad de función – Externo
Nombre en inglés: Function understandability
Descripción: ¿Cuál es la proporción de funciones producto el usuario podrá entender correctamente? realizar pruebas de conducta de usuario y entrevista de usuario con cuestionarios. Cuenta el número de funciones de interfaz de usuario donde el propósito es la facilidad de entendimiento por el usuario y compara con el número de funciones disponibles para el usuario

Fórmula:

$$X=A/B,$$

A (conteo): Número de funciones interfaz cuyo propósito es descrito correctamente por el usuario.

B (conteo): Número funciones disponibles desde la interfaz.

Atributo al que ayuda a calcular: Comprensibilidad

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 69

Nombre en español: Entrada y salida entendible

Nombre en inglés: Understandable input and output

Descripción: ¿Pueden los usuarios entender qué es requerido como dato de entrada y qué provee como salida por el sistema software? Se realiza pruebas de conducta de usuario y entrevista de usuario con cuestionarios u observación de comportamiento. Cuenta el número de datos de entradas y salidas entendidas por el usuario y compara con el número total disponible para el usuario.

Fórmula:

$$X=A/B,$$

A (conteo): Número de datos de entrada y salida que el usuario entiende con éxito.

B (conteo): número de datos de entrada y salida disponibles para la interfaz.

Atributo al que ayuda a calcular: Comprensibilidad

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 70

Nombre en español: Completitud de descripción- Interno

Nombre en inglés: Completeness of description

Descripción: ¿Qué proporción de funciones (o tipos de funciones) son descritas en la descripción del producto? Cuenta el número de funciones las cuales son adecuadamente descritas y compara con el número total de funciones que requieren capacidad de demostración

Fórmula:

$$X=A/B,$$

A (conteo): Número de funciones (o tipos de funciones) descritas en la descripción del producto.

B (conteo): número total de funciones(o tipos de funciones).

Atributo al que ayuda a calcular: Comprensibilidad

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 mejor.

Referencias: ISO 9126 [15]: 3

Espectro: Gris

MÉTRICA 71

Nombre en español: Capacidad de demostración

Nombre en inglés: Demonstration capability

Descripción: ¿Qué proporción de funciones que requieren demostración tienen capacidad de demostración? Cuenta el número de funciones que son adecuadamente demostrables y compara con el número total de funciones que requieren capacidad de demostración

Fórmula:

$$X=A/B,$$

A (conteo): Número de funciones demostradas y confirmadas en revisión.

B (conteo): número total de funciones que requieren capacidad de demostración.

Atributo al que ayuda a calcular: Comprensibilidad

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 más capacidad.

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 72

Nombre en español: Funciones evidentes – Interno

Nombre en inglés: Evident functions

Descripción: ¿Qué proporción de funciones del producto son evidentes para el usuario? Cuenta el número de funciones que son evidentes para el usuario y compara con el número total de funciones

Fórmula:

$$X=A/B,$$

A (conteo): Número de funciones(o tipos de funciones) evidentes para el usuario.

B (conteo): número total de funciones(o tipos de funciones).

Atributo al que ayuda a calcular: Comprensibilidad

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 mejor

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 73

Nombre en español: Comprensibilidad de función – Interno

Nombre en inglés: Function understandability

Descripción: ¿Qué proporción de funciones del producto será capaz el usuario de entender correctamente? Cuenta el número de funciones de interfaz de usuario donde el propósito es entendido por el usuario y compara con el número total de funciones de interfaz de usuario

Fórmula:

$$X=A/B,$$

A (conteo): Número de funciones de interfaz de usuario cuyo propósito es entendido por el usuario.

B (conteo): número funciones de interface de usuario

Atributo al que ayuda a calcular: Comprensibilidad

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 mejor.

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 74

Nombre en español: Concisión 1

Nombre en inglés: Conciseness 1

Descripción: ¿El código es accesible?
Atributo al que ayuda a calcular: Concisión.
Referencias: Boehm [17]
Espectro: Oscuro
Otros: Puede estar relacionado con el atributo de accesibilidad

MÉTRICA 75

Nombre en español: Concisión 2
Nombre en inglés: Conciseness 2
Descripción: ¿Son conservados fuera de los ciclos todos los cómputos no dependientes de ellos?
Atributo al que ayuda a calcular: Concisión
Referencias: Boehm [17]
Espectro: Oscuro

MÉTRICA 76

Nombre en español: Concisión 3
Nombre en inglés: Conciseness 3
Descripción: ¿Evitan expresiones compuestas repetidas en sentencias aritméticas?
Fórmula:
Atributo al que ayuda a calcular: Concisión
Referencias: Boehm [17]
Espectro: Oscuro

MÉTRICA 77

Nombre en español: Concisión 4
Nombre en inglés: Conciseness 4
Descripción: ¿Los comentarios descriptivos asociados tienen un bajo “índice de niebla”? (Referencia: R. Gunning, pag 6-4) (sección 6.0)
Atributo al que ayuda a calcular: Concisión
Referencias: Boehm [17]
Espectro: Oscuro

MÉTRICA 78

Nombre en español: Concisión 5
Nombre en inglés: Conciseness 5
Descripción: ¿Existe una transferencia a todas las sentencias etiquetadas y es fácilmente localizada la transferencia?
Atributo al que ayuda a calcular: Concisión
Referencias: Boehm [17]
Espectro: Oscuro

MÉTRICA 79

Nombre en español: Concisión 6

Nombre en inglés: Conciseness 6

Descripción: ¿Está el programa sin arreglos multidimensionales cualquiera para los cuales bastarían unos pocos arreglos con poca dimensión?

Atributo al que ayuda a calcular: Concisión

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 80

Nombre en español: Concisión 7

Nombre en inglés: Conciseness 7

Descripción: ¿evitan los formatos especiales o entradas/salidas innecesarias? De no ser así, ¿podría ser usado algún otro formato existente en su lugar sin pérdida de información (e.g., FORMAT (I10) y FORMAT (2I10))?

Atributo al que ayuda a calcular: Concisión

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 81

Nombre en español: Concisión 8

Nombre en inglés: Conciseness 8

Descripción: ¿Son usados términos de decisión binarios cuando expresiones de bifurcación pueden ser evaluadas sólo como “verdaderas” ó “falsas”?

Atributo al que ayuda a calcular: Concisión

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 82

Nombre en español: Métricas de Hastead

Nombre en inglés: Halstead Metrics

Descripción: Basa su medición en 5 métricas para calcular el esfuerzo. Hace uso de los siguientes elementos:

μ_1 : Número de operadores únicos,

μ_2 : Número de operandos únicos,

N_1 : Ocurrencias totales de los operadores,

N_2 : Ocurrencias totales de los operandos. Siendo el programa P

Fórmulas:

1. Longitud: $N(P) \rightarrow N = N_1 + N_2$.

2. Vocabulario: $\mu(P) \rightarrow \mu = \mu_1 + \mu_2$.

3. Volumen: $V(P) \rightarrow V = N * \log \mu$.

4. Dificultad del programa: $D(P) \rightarrow D = (\mu_1/2) * (N_2/\mu_2)$

5. Esfuerzo: Para generar P $\rightarrow E = D * V$

Atributo al que ayuda a calcular: Concisión.

Referencias: Andersson [74], Kan [24], LAIRD [76], McCall [18]-1, Choque [89], Gaffney [63]

Espectro: Gris

Otros: Aplica en disciplinas de análisis, diseño, implementación a nivel de sistema.es en general una métrica de código fuente

MÉTRICA 83

Nombre en español: Conformidad funcional- Externo

Nombre en inglés: Functional compliance

Descripción: ¿Cuan compatible es la funcionalidad del producto para regulaciones, estándares y convenciones aplicables? Cuenta el número de ítems encontrados que requieren cumplimiento y compara con el número de ítems que requieren cumplimiento tal y como están en la especificación. Diseña casos de prueba de acuerdo con los ítems de conformidad. Dirigir pruebas funcionales para esos casos de prueba. Cuenta el número de ítems conformes que han sido satisfechos

Fórmula:

$$X=1 - A/B,$$

A (conteo): Número de ítems específicos conformes a la Funcionalidad que no han sido implementados durante las pruebas.

B (conteo): número total de ítems específicos conformes a la Funcionalidad

Atributo al que ayuda a calcular: Conformidad con la Funcionalidad (Conformidad)

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 84

Nombre en español: Conformidad con interface estándar

Nombre en inglés: Interface estándar compliance

Descripción: ¿Cuan compatibles son las interfaces para regulaciones, estándares y convenciones aplicables? Cuenta el número de interfaces que encuentra necesaria la conformidad y compara con el número de interfaces que requieren conformidad como en las especificaciones. NOTA: todos los atributos especificados de un estándar deben ser probados

Fórmula:

$$X=A/B,$$

A (conteo): Número de interfaces especificadas implementados correctamente.

B (conteo): número total de interfaces que requieren conformidad.

Atributo al que ayuda a calcular: Conformidad con la Funcionalidad (Conformidad)

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 85

Nombre en español: Conformidad funcional – interno

Nombre en inglés: Functional compliance

Descripción: ¿Cuan compatible es la funcionalidad del producto para regulaciones, estándares y convenciones aplicables? Cuenta el número de ítems encontrados que requieren cumplimiento y compara con el número de ítems que requieren cumplimiento como en la especificación

Fórmula:

$$X=A/B,$$

A (conteo): Número de ítems implementados correctamente relacionados a Conformidad con Funcionalidad confirmada en evaluación.

B (conteo): número funciones de interface de usuario.

Atributo al que ayuda a calcular: Conformidad con la Funcionalidad (Conformidad)

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 más compatible.

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 86

Nombre en español: Conformidad estándar inter sistema

Nombre en inglés: Intersystem estándar compliance

Descripción: ¿Cuan compatibles son las interfaces para regulaciones, estándares y convenciones aplicables? Cuenta el número de interfaces encontradas que requieren cumplimiento y compara con el número de ítems que requieren cumplimiento como en la especificación.

Fórmula:

$$X=A/B,$$

A (conteo): Número de interfaces implementados correctamente como en las especificaciones.

B (conteo): número total de interfaces necesitando Conformidad.

Atributo al que ayuda a calcular: Conformidad con la Funcionalidad (Conformidad)

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 más compatible.

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 87

Nombre en español: Conformidad con la Confiabilidad- Externo

Nombre en inglés: Reliability compliance

Descripción: ¿Cuan compatible es la Confiabilidad del producto para regulaciones, estándares y convenciones aplicables? Cuenta el número de

ítems encontrados que requieren cumplimiento y compara con el número de ítems que requieren cumplimiento como en la especificación

Fórmula:

$$X = 1 - A/B,$$

A (conteo): Número de ítems especificados conformes a la Confiabilidad que no han sido implementados durante las pruebas.

B (conteo): número total de ítems especificados conformes a la Confiabilidad.

Atributo al que ayuda a calcular: Conformidad con la Confiabilidad (Conformidad)

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 88

Nombre en español: Conformidad con la Confiabilidad- interno

Nombre en inglés: Reliability compliance

Descripción: ¿Cuan compatible es la confiabilidad del producto para regulaciones, estándares y convenciones aplicables? Cuenta el número de ítems encontrados que requieren cumplimiento y compara con el número de ítems que requieren cumplimiento como en la especificación

Fórmula:

$$X = 1 - A/B,$$

A (conteo): Número de ítems establecidos conformes a la Usabilidad que no han sido implementados durante las pruebas.

B (conteo): Número total de ítems especificados conformes a la Usabilidad.

Atributo al que ayuda a calcular: Conformidad con la Confiabilidad (Conformidad)

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 mejor.

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 89

Nombre en español: Conformidad con la Usabilidad - externo

Nombre en inglés: Reliability compliance

Descripción: ¿Qué tan completamente el software es adherido a los estándares, convenciones, guías de estilo o reglamentos relacionados a Usabilidad? Especifica cumplimiento de productos basados en estándares, convenios, guías de estilo o reglamentos relacionados a Usabilidad. Diseña casos de prueba de acuerdo con ítems de Conformidad. Realizar pruebas funcionales para estos casos de prueba.

Fórmula:

$$X = 1 - A/B,$$

A (conteo): Número de ítems establecidos conformes a la Usabilidad que no han sido implementados durante las pruebas.

B (conteo): Número total de ítems especificados conformes a la Usabilidad.

Atributo al que ayuda a calcular: Conformidad con la Usabilidad (Conformidad)

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 90

Nombre en español: Conformidad con la Usabilidad – interno

Nombre en inglés: Usability compliance

Descripción: ¿Cuan compatible es el producto para regulaciones, estándares y convenciones aplicables para Usabilidad? Cuenta el número de ítems encontrados que requieren cumplimiento y compara con el número de ítems que requieren cumplimiento como en la especificación

Fórmula:

$$X=A/B,$$

A (conteo): Número de ítems implementados correctamente relacionados a Usabilidad confirmados en evaluación.

B (conteo): número total de ítems compatibles.

Atributo al que ayuda a calcular: Conformidad con la Usabilidad (Conformidad)

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 más compatible.

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 91

Nombre en español: Conformidad con la Eficiencia – Externo

Nombre en inglés: Efficiency compliance

Descripción: ¿Qué tan adaptable es la Eficiencia al producto para aplicar reglamentación, estándares y convenciones? Cuenta el número de ítems encontrados que requieren cumplimiento y compara con el número de ítems que requieren cumplimiento como en la especificación

Fórmula:

$$X= 1 - A/B,$$

X: proporción de ítems conformes a la satisfacción relacionados a la Eficiencia),

A (conteo): Número de ítems establecidos conformes a la Eficiencia que no han sido implementados durante las pruebas.

B (conteo): número total de ítems especificados conformes a la Eficiencia. NOTA: esta métrica puede ser útil para recoger varios valores de medidas en el tiempo, analizar la tendencia de incremento satisfecha en Conformidad a los ítems y determinar si totalmente satisfechos o no.

Atributo al que ayuda a calcular: Conformidad con la Eficiencia (Conformidad)

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 92

Nombre en español: Conformidad con la Eficiencia - Interno

Nombre en inglés: Efficiency compliance

Descripción: ¿Cuan compatible es la Eficiencia del producto para regulaciones, estándares y convenciones aplicables? Cuenta el número de ítems encontrados que requieren cumplimiento y compara con el número de ítems que requieren cumplimiento como en la especificación

Fórmula:

$$X=A/B,$$

A (conteo): Número de ítems implementados correctamente relacionados a Eficiencia confirmados en evaluación.

B (conteo): número total de ítems compatibles.

Atributo al que ayuda a calcular: Conformidad Eficiencia (Conformidad)

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 más compatible.

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 93

Nombre en español: Conformidad con la Mantenibilidad – Externo

Nombre en inglés: Maintainability compliance

Descripción: ¿Cuan compatible es la Mantenibilidad del producto para regulaciones, estándares y convenciones aplicables? Cuenta el número de ítems encontrados que requieren cumplimiento y compara con el número de ítems que requieren cumplimiento como en la especificación

Fórmula:

$$X= 1 - A/B,$$

A (conteo): Número de ítems establecidos conformes a la Mantenibilidad que no han sido implementados durante las pruebas.

B (conteo): número total de ítems especificados conformes a la Mantenibilidad.

Atributo al que ayuda a calcular: Conformidad con la Mantenibilidad (Conformidad)

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 94

Nombre en español: Conformidad con la Mantenibilidad – Interno

Nombre en inglés: Maintainability compliance

Descripción: ¿Cuan compatible es la Mantenibilidad del producto para regulaciones, estándares y convenciones aplicables? Cuenta el número de

ítems encontrados que requieren cumplimiento y compara con el número de ítems que requieren cumplimiento como en la especificación

Fórmula:

$$X=A/B,$$

A (conteo): Número de ítems implementados correctamente relacionados a Mantenibilidad confirmados en evaluación.

B (conteo): número total de ítems compatibles.

Atributo al que ayuda a calcular: Conformidad con la Mantenibilidad (Conformidad)

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 más compatible.

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 95

Nombre en español: Conformidad con la Portabilidad- Externo

Nombre en inglés: Portability compliace

Descripción: ¿Cuan compatible es la Portabilidad del producto para regulaciones, estándares y convenciones aplicables? Cuenta el número de ítems encontrados que requieren cumplimiento y compara con el número de ítems que requieren cumplimiento como en la especificación

Fórmula:

$$X= 1 - A/B,$$

A (conteo): Número de ítems establecidos conformes a la Portabilidad que no han sido implementados durante las pruebas.

B (conteo): número total de ítems especificados conformes a la Portabilidad.

Atributo al que ayuda a calcular: Conformidad Portabilidad (Conformidad)

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 96

Nombre en español: Conformidad con la Portabilidad – Interno

Nombre en inglés: Portability compliace

Descripción: ¿Cuan compatible es la Portabilidad del producto para regulaciones, estándares y convenciones aplicables? Cuenta el número de ítems encontrados que requieren cumplimiento y compara con el número de ítems que requieren cumplimiento como en la especificación

Fórmula:

$$X=A/B,$$

A (conteo): Número de ítems implementados correctamente relacionados a Portabilidad confirmados en evaluación.

B (conteo): número total de ítems compatibles.

Atributo al que ayuda a calcular: Conformidad con la Portabilidad (Conformidad)

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 más completo.

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 97

Nombre en español: Consistencia 1

Nombre en inglés: Consistency 1

Descripción: ¿Son todas las especificaciones de conjuntos de variables globales (i.e., aquellas que aparecen en dos o más subprogramas) idénticas (e.g., rotuladas COMMON)?

Atributo al que ayuda a calcular: Consistencia

Referencias: Boehm [17]

Espectro: Oscura

MÉTRICA 98

Nombre en español: Consistencia 2

Nombre en inglés: Consistency 2

Descripción: ¿Es el tipo de variable (e.g., real, entero, etc.) consistente para todos los usos?

Atributo al que ayuda a calcular: Consistencia

Referencias: Boehm [17]

Espectro: Oscura

MÉTRICA 99

Nombre en español: Consistencia 3

Nombre en inglés: Consistency 3

Descripción: ¿Son incluidos todos los subíndices en todos los usos de arrays dimensionados?

Atributo al que ayuda a calcular: Consistencia

Referencias: Boehm [17]

Espectro: Oscura

MÉTRICA 100

Nombre en español: Consistencia 4

Nombre en inglés: Consistency 4

Descripción: ¿Aparecen todos los parámetros de subíndices en una sentencia de especificación que identifique los parámetros como un array dimensionado?

Atributo al que ayuda a calcular: Consistencia

Referencias: Boehm [17]

Espectro: Oscura

MÉTRICA 101

Nombre en español: Consistencia 5

Nombre en inglés: Consistency 5

Descripción: ¿El programa contiene solamente una representación para constantes físicas o matemáticas (e.g., no 3.14159 en un lugar y 3.1416 en otro)?

Atributo al que ayuda a calcular: Consistencia

Referencias: Boehm [17]

Espectro: Oscura

MÉTRICA 102

Nombre en español: Consistencia 6

Nombre en inglés: Consistency 6

Descripción: ¿Son consistentes la tolerancia sobre iteración, integración, etc., con el número de dígitos significativos en las entradas y salidas?

Atributo al que ayuda a calcular: Consistencia

Referencias: Boehm [17]

Espectro: Oscura

MÉTRICA 103

Nombre en español: Consistencia 7

Nombre en inglés: Consistency 7

Descripción: ¿Es consistente el uso de la función con sus tipos definidos?

Atributo al que ayuda a calcular: Consistencia

Referencias: Boehm [17]

Espectro: Oscura

MÉTRICA 104

Nombre en español: Consistencia 8

Nombre en inglés: Consistency 8

Descripción: ¿Es usada siempre la misma marca (descriptor nemónico) para identificar la misma salida del programa?

Atributo al que ayuda a calcular: Consistencia

Referencias: Boehm [17]

Espectro: Oscura

MÉTRICA 105

Nombre en español: Consistencia 9

Nombre en inglés: Consistency 9

Descripción: ¿Son funcionalmente similares las expresiones aritméticas similarmente construidas? (e.g., $Y = \text{SQRT}(x)$ en un lugar; $Z = Q^{**}.5$ en otro)

Atributo al que ayuda a calcular: Consistencia

Referencias: Boehm [17]

Espectro: Oscura

MÉTRICA 106

Nombre en español: Consistencia 10

Nombre en inglés: Consistency 10

Descripción: ¿Es usado un nombre de variable para representar diferentes entidades físicas en el programa? ¿Son representadas diferentes entidades físicas por diferentes nombres en el programa, en lugar de un nombre de variable?

Atributo al que ayuda a calcular: Consistencia

Referencias: Boehm [17]

Espectro: Oscura

MÉTRICA 107

Nombre en español: Consistencia 11

Nombre en inglés: Consistency 11

Descripción: ¿Es la misma entidad física representada por un nombre, en lugar de dos o más nombres de variables?

Atributo al que ayuda a calcular: Consistencia

Referencias: Boehm [17]

Espectro: Oscura

MÉTRICA 108

Nombre en español: Consistencia 12

Nombre en inglés: Consistency 12

Descripción: ¿Están todos los elementos de un array funcionalmente relacionados?

Atributo al que ayuda a calcular: Consistencia

Referencias: Boehm [17]

Espectro: Oscura

MÉTRICA 109

Nombre en español: Coherencia de procedimiento

Nombre en inglés: Procedure consistency (design and implementation at system level)

Descripción: La métrica es la suma de los puntajes de los siguientes elementos aplicables dividido el número de elementos aplicables:

1. Representación de diseño estándar: Diagramas de flujo, diagramas HIPO, Lenguaje de diseño del programa - desde cualquier representación de diseño usada, estándares para representar los elementos de control de flujo deben establecerse y seguir. estos elementos aplican a diseño solamente. la medida está basada en el número de módulos cuya representación de diseño no obedece con estándares

2. Convenciones de secuencia de llamado: Interacciones entre módulos deben ser estandarizadas. los estándares deben ser establecidos durante el diseño y seguidos durante la implementación. la medida es basada en el número de módulos que no cumplen con las convenciones
3. Convenciones Entrada/Salida: Convenciones para las cuales los módulos realizaran I/O, que serán completados, y los formatos de I/O son establecidos y seguidos. la medida está basada sobre que módulos no cumplen con las convenciones
4. Convenciones de manejo e error: Un método consistente para manejo de error es requerido. Las convenciones establecidas en diseño son seguidas en implementación. la medida está basada en el número de módulos que no cumplen con las convenciones

Fórmula: A nivel de sistema:

$$X=A/B,$$

$A=\sum a_i$, (Para i de 1 a 4)

a_i : puntajes de los elementos aplicables,

B (conteo): Número de elementos aplicables.

Para cualquier elemento aplicable (1), (2), (3) o (4) usar la formula

$$Y= (1 - C/D),$$

C (Conteo): Número de módulos que violan la regla,

D (Conteo): Número total de módulos

Atributo al que ayuda a calcular: Consistencia

Granularidad: Sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica en disciplinas de diseño, implementación

MÉTRICA 110

Nombre en español: Consistencia de datos

Nombre en inglés: Data consistency

Descripción: La métrica es la suma de los puntajes de los siguientes elementos aplicables dividido el número de elementos aplicables:

1. Representación de uso del dato estándar: en coordinación con CS1. una representación de diseño estándar para uso de datos es establecido y seguido. esta es solo una métrica de diseño, identificando el número de módulos que violan los estándares
2. Convenciones de nombrado: convenciones de nombrado para variables y módulos son establecidas y seguidas.
3. Consistencia de unidad: unidades de variables son escogidas consistentes con todos los usos de variables. la medida está basada en

el número de módulos que las unidades de consistencia no son utilizadas. esta puede ser medida en ambos diseño e implementación

4. Definiciones de consistencia global: los elementos de datos globales son definidos en la misma forma por todos los módulos. la medida está basada en el número de módulos en que los elementos de datos globales son definidos en una forma inconsistente para ambos diseño e implementación.
5. Consistencia de tipo de dato: un elemento de dato definido como un tipo de dato particular es usado como ese tipo de dato en todas las ocurrencias. una violación común de esta regla es encontrada en arreglos donde varios tipos de datos son definidos. la medida está basada en el número de módulos que utilizan tipos de datos inconsistentemente

Fórmula: A nivel de sistema:

$$X=A/B,$$

A = $\sum a_i$, (Para i de 1 a 4)

a_i : puntajes de los elementos aplicables,

B: Número de elementos aplicables.

Para cualquier elemento aplicable (1), (2), (3), (4) o (5) usar la formula

$$Y = (1 - C/D),$$

C: Número de módulos que violan la regla,

D: Número total de módulos

Atributo al que ayuda a calcular: Consistencia

Granularidad: Sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica en disciplinas de diseño, implementación

MÉTRICA 111

Nombre en español: Métrica de acceso de datos

Nombre en inglés: Data access metric (DAM)

Descripción: Esta métrica es el cociente del número de atributos privados (protegidos) para el número total de atributos declarados en la clase. Un alto valor de DAM es deseable. Rango entre 0 y 1,

Fórmula:

$$DAM = A/B$$

A: Número de atributos privados

B: número total de atributos de la clase

Atributo al que ayuda a calcular: Encapsulamiento

Influencia sobre otros atributos: Flexibilidad, Comprensibilidad, Efectividad

Granularidad: Atributo

Referencias: Bansiya [31], JETTER [3], OLMEDILLA [53], Garzas [57]

Espectro: Gris

MÉTRICA 112

Nombre en español: Proporción de métodos ocultos

Nombre en inglés: Method hiding factor- MHF

Descripción: La métrica es la proporción entre la suma de los grados de invisibilidad de los métodos en todas las clases y el número total de métodos definidos en el sistema. MHF es la proporción entre los métodos definidos como protegidos o privados y el número total de métodos. MHF es propuesta como una medida de Encapsulamiento, cantidad relativa de información oculta. Cuando incrementa MHF, la densidad de defectos y el esfuerzo necesario para corregirlos debería disminuir. Los métodos heredados no son considerados.

Fórmula:

$$MHF = \frac{\sum_{i=1}^{TC} \text{de } [\sum_{m=1}^{Md(C_i)} (C_i) \text{ de } (1-V(M_{mi}))]}{\sum_{i=1}^{TC} \text{de } [Md(C_i)]}$$

Donde:

$$V(M_{mi}) = \frac{\sum_{j=1}^{TC} \text{de } [es_visible(M_{mi}, C_j)]}{TC}$$

es_ visible (Mmi, Cj) = (1 si y solo si Cj puede llamar a Mmi, 0 en caso contrario).

TC: es el número total de clases,

Md (Cj): número de métodos definidos en la clase Ci (no heredados),

V (Mmi): es la visibilidad: porcentaje total de clases desde las cuales el método Mmi es visible.

Atributo al que ayuda a calcular: Encapsulamiento

Granularidad: Clase

Referencias: VAZQUEZ [61, Abreu y Melo [79], PRESSMAN [1], Champeaux [80], Marin [28]

Espectro: Claro

Otros: Aplica en disciplina de implementación

MÉTRICA 113

Nombre en español: Proporción de atributos ocultos

Nombre en inglés: Attribute hiding factor- AHF

Descripción: La métrica es la proporción entre la suma de los grados de invisibilidad de los atributos en todas las clases y el número total de atributos definidos en el sistema. AHF es la proporción entre los atributos definidos como protegidos o privados y el número total de atributos. Idealmente el valor de esta métrica debería ser siempre el 100%, intentando ocultar todos los atributos. Las pautas de diseño sugieren que no hay que emplear atributos públicos, ya que considera que esto viola los principios de Encapsulamiento al exponer la implementación de las clases. Para mejorar el rendimiento, a veces evitar el uso de métodos que acceden o modifican atributos (métodos get/set)

accediendo a ellos directamente. En esta práctica debe extremarse la prudencia y evaluar si realmente los pros son mayores que los contras

Fórmula:

$$AHF = \frac{\{\sum \text{ para } i=1 \text{ hasta } TC \text{ de } [\sum \text{ para } m=1 \text{ hasta } Ad(C_i) \text{ de } (1-V(A_{mi}))]\}}{\{\sum \text{ para } i=1 \text{ hasta } TC \text{ de } [Ad(C_i)]\}}$$

Donde:

$$V(A_{mi}) = \frac{\{\sum \text{ para } j=1 \text{ hasta } TC \text{ de } [es_visible(A_{mi}, C_j)]\}}{TC}$$

es_visible (A_{mi}, C_j) = (1 si y solo si C_j puede llamar a A_{mi}, 0 en caso contrario).

TC: es el número total de clases,

Ad(C_j): número de atributos definidos en la clase C_i (no heredados),

V(A_{mi}): es la visibilidad: porcentaje total de clases desde las cuales el atributo A_{mi} es visible.

Atributo al que ayuda a calcular: Encapsulamiento

Granularidad: Clase

Referencias: VAZQUEZ [61], Abreu y Melo [79], PRESSMAN [1], Champeaux [80], Marin [28]

Espectro: Claro

Otros: Aplica en disciplina de implementación

MÉTRICA 114

Nombre en español: Carencia de ambigüedad

Nombre en inglés: lack ambiguity

Descripción: Ausencia de ambigüedad de los requisitos funcionales y no funcionales

Fórmula:

$$Q1 = n_{ui} / n_r$$

n_r: número de requisitos para los que todos los revisores tuvieron interpretaciones idénticas.

n_r = n_f + n_{nf},

n_r: número de requisitos de la especificación

n_f: Número de requisitos funcionales,

n_{nf}: número de requisitos no funcionales. Cuanto más cerca a 1 este

Q menor será la ambigüedad de la interpretación

Atributo al que ayuda a calcular: Especificado

Referencias: PRESSMAN [1]

Espectro: Oscuro

MÉTRICA 115

Nombre en español: Exactitud 1

Nombre en inglés: Accuracy 1

Descripción: ¿Son usados por el programa métodos numéricos suficientemente adecuados para una aplicación específica?

Atributo al que ayuda a calcular: Exactitud

Referencias: Boehm [17]
Espectro: Oscuro

MÉTRICA 116

Nombre en español: Exactitud de expectativa

Nombre en inglés: Accuracy to expectation

Descripción: ¿Hay diferencias entre los resultados actuales y los aceptables esperados razonables? Hacer casos de pruebas de entrada vs salida y comparar con las salidas para resultados esperados razonables. Cuenta el número de casos encontrados por los usuarios con una diferencia inaceptable para resultados esperados razonables.

Fórmula:

$$X=A/T,$$

A (Conteo)=Número de casos encontrados por los usuarios con una diferencia en contra de resultados esperados razonables mas allá de lo permitido.

T (tiempo)= tiempo de operación.

Atributo al que ayuda a calcular: Exactitud

Rango: $0 \leq X$, entre más cerca a 0 mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 117

Nombre en español: Exactitud computacional- externo

Nombre en inglés: Computational Accuracy

Descripción: ¿Qué tan seguido los usuarios finales encuentran resultados inexactos? Registra el número de cálculos inexactos basados en las especificaciones.

Fórmula:

$$X=A/T,$$

A (Cuenta)=Número cálculos inexactos encontrados por usuarios.

T (tiempo)= tiempo de operación.

Atributo al que ayuda a calcular: Exactitud

Rango: $0 \leq X$, entre más cerca a 0 mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 118

Nombre en español: Precisión - externo

Nombre en inglés: Precision

Descripción: ¿Qué tan seguido los usuarios finales encuentran resultados con precisión deficiente? Registre el número de resultados con precisión deficiente.

Fórmula:

$$X=A/T,$$

A (Conteo)=Número de resultados encontrados por usuarios con un nivel de precisión diferente de los requerimientos.

T (tiempo)= tiempo de operación. $0 \leq X$, entre más cerca a 0 mejor

Atributo al que ayuda a calcular: Exactitud

Rango: $0 \leq X$, entre más cerca a 0 mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 119

Nombre en español: Exactitud computacional- interno

Nombre en inglés: Computational Accuracy

Descripción: ¿Qué tan completamente los requerimientos de exactitud han sido aplicados? Cuenta el número de funciones que tienen implementados los requerimientos de exactitud y compara con el número de funciones con requerimientos de exactitud especificados.

Fórmula:

$$X=A/B,$$

A (Conteo): Número de funciones en que los requerimientos de exactitud específicos han sido implementados, confirmados como en la evacuación,

B (Conteo): Número de funcione para las cuales los requerimientos de exactitud específicos necesitan ser implementados.

Atributo al que ayuda a calcular: exactitud

Rango: $0 \leq X \leq 1$, entre más cerca a 1, más completo

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 120

Nombre en español: Precisión- interno

Nombre en inglés: Precision

Descripción: ¿Qué tan completa fue la implementación de niveles específicos de precisión para los datos? Cuenta el número de los datos que son implementados con niveles específicos de precisión y compara al número total de datos con un nivel específico de requerimientos de precisión

Fórmula:

$$X=A/B,$$

A (Conteo): Número de datos implementados con niveles específicos de precisión, confirmados como en la evaluación,

B (Conteo): Número de datos que requieren niveles específicos de precisión.

Atributo al que ayuda a calcular: Exactitud

Rango: $0 \leq X \leq 1$, entre más cerca a 1, mas completo

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 121

Nombre en español: Lista de chequeo de precisión

Nombre en inglés: Accuracy checklist

Descripción: Cada elemento es una medida binaria indicando existencia, o ausencia de los elementos. La métrica es la suma de puntajes de los siguientes elementos aplicables divididos por el número de elementos aplicables:

1. Análisis de error ejecutado y presupuestado para el modulo: (solo fase de requerimientos) Un análisis de error debe ser parte de el análisis de requerimientos realizado para desarrollar la especificación de requerimientos. este análisis asigna requerimientos de exactitud general a funciones individuales a ser realizadas por el sistema. este presupuesto requerimientos de exactitud provee objetivos definitivos para los diseñadores e implementadores de modulo.
2. Una declaración definitiva de requerimientos para exactitud de entradas, salidas, procesamiento y constantes:(solo fase de requerimientos) igual a la explicación de (1).
3. Suficiencia de librería matemática:(solo fase de diseño) la exactitud de las rutinas de librerías matemáticas utilizadas dentro del sistema es comprobada para consistencia con los objetivos de exactitud general
4. Suficiencia de métodos numéricos:(fases de diseño e implementación) los métodos numéricos utilizados dentro del sistema son consistentes con los objetivos de exactitud. ellos pueden ser comprobados en diseño e implementación
5. Ejecución de salida con tolerancias: (solo en la fase de implementación requiriendo ejecución) una medida final durante pruebas de desarrollo es la ejecución de módulos y comprobación para exactitud de salidas.

Fórmula: A nivel de sistema:

$$X=A/B,$$

A: Puntaje total de los elementos aplicables,

B: Número de elementos aplicables

Atributo al que ayuda a calcular: Exactitud

Granularidad: sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica disciplinas de análisis, diseño, implementación en el nivel del sistema

MÉTRICA 122

Nombre en español: Medida de expansión de almacenamiento de los datos

Nombre en inglés: Data storage expansión measure

Descripción: La métrica es la suma de los puntajes de los siguientes elementos aplicables dividido por el número de elementos aplicables:

1. Procesamiento lógico independiente de especificación/requerimientos de almacenamiento (por modulo): el procesamiento lógico de un modulo es independiente del tamaño de almacenamiento, espacio de buffer, o tamaños de arreglos. el diseño provee para dimensiones variables y tamaño de arreglos dinámicos definidos para métrica mente. la métrica esta basada en el número de módulos que contienen dimensiones de código difícil que no ejemplifican este concepto
2. Porcentaje de capacidad de memoria no comprometida:(Solo implementación) La cantidad de memoria disponible para la expansión es una medida importante. Esta medida determina el porcentaje de memoria disponible que no ha sido utilizada en la implementación del sistema actual

Fórmula: A nivel de sistema:

$$X=A/B,$$

A= puntaje total de los elementos aplicables,
B (conteo): Número de elementos aplicables.

ELEMENTOS:

- (1): $Y=(1-C/D)$,
C: Número de módulos que violan la regla,
D: Número total de módulos,
- (2): $Y=(1-C/D)$,
C (Conteo): Cantidad de memoria no comprometida,
D (Conteo): total de memoria disponible

Atributo al que ayuda a calcular: Expansibilidad

Granularidad: Sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica en las disciplinas de diseño e implementación a nivel de sistema

MÉTRICA 123

Nombre en español: Medida de Extensibilidad

Nombre en inglés: Extensibility Measure

Descripción: La métrica es la suma de los puntajes de los siguientes elementos aplicables dividido por el número de elementos aplicables:

- (1) Exactitud, convergencia, calendario de control de procesamiento de los atributos que son paramétricos: Un módulo que pueden proporcionar diferentes grados de convergencia o el calendario para lograr una mayor precisión provee este atributo de Extensibilidad. parámetros de control de Hard-code, contadores, valores de reloj, etc. violan esta medida. Esta medida esta basada en el número de módulos que no ejemplifican esta

característica. una determinación puede hacerse durante el diseño e implementación.

- (2) Impulso de módulos de tabla: El uso de tablas dentro de un módulo facilita distintas representaciones y características de procesamiento. Esta medida que puede aplicarse durante el diseño y la implementación esta basada en el número de módulos que no están impulsados por la tabla.
- (3) Porcentaje de la velocidad sin comprometer la capacidad: (solo la implementación) Una cierta función puede ser necesaria en la especificación de requisitos de desempeño a ser realizado en un período de tiempo especificado en el calendario general de objetivos. La cantidad de tiempo no utilizado por la actual aplicación de la función es el tiempo disponible de procesamiento para la posible expansión de la capacidad computacional. que identifica el por ciento del total de tiempo de procesamiento que la capacidad no compromete calendario ni objetivos. La cantidad de tiempo no utilizado por la actual aplicación de la función es el tiempo de procesamiento $av,-h$) el potencial para la expansión de la capacidad computacional. Identifica el seguro por ciento del total de tiempo de procesamientos no comprometidos Thal

Fórmula: A nivel de sistema:

$$X=A/B,$$

A= Puntaje total de los elementos aplicables,

B: Número de elemetos aplicables.

ELEMENTOS:

- (1): $Y=(1-C/D)$,
C: Número de módulos que violan la regla,
D: Número total de módulos
- (2): $Y=(1-C/D)$,
C: Número de módulos que no impulsa la tabla,
D: Número total de módulos,
- (3): $Y= C/D$,
C: Cantidad de tiempo del ciclo no comprometido,
D: Tiempo total de procesamiento

Atributo al que ayuda a calcular: expansibilidad

Granularidad: Sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica en las disciplinas de diseño e implementación a nivel de sistema

MÉTRICA 124

Nombre en español: Grado en que un modulo es referenciado por otros módulos

Nombre en inglés: Extent to which module is referenced by other modules (References)

Descripción: Esta métrica proporciona una medida de la generalidad de los módulos que utilicen en el sistema actual. Un módulo que es considerado de carácter más general, si es utilizado (referencia) por más de un módulo. El número de estos módulos comunes dividido por el número total de módulos que ofrece la medida.

Fórmula:

$$X=A/B,$$

A: Número de módulos comunes,

B: Número total de módulos

Atributo al que ayuda a calcular: Generalidad

Granularidad: Sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica en las disciplinas de diseño e implementación a nivel de sistema

MÉTRICA 125

Nombre en español: Implementación para la lista de chequeo de generalidad

Nombre en inglés: Implemtation for generality checklist (Implementation generality)

Descripción: Esta métrica es la suma de los puntajes de los siguientes elementos aplicables, dividido por el número de elementos aplicables.

- (1) Funciones de Entrada, procesamiento, salida no son mezcladas en una única función: Un modulo que realiza E / S, así como procesamiento no es tan general como un módulo que simplemente realiza el procesamiento. Esta medida eata basada en el número de módulos que violan este concepto en diseño e implementación.
- (2) Aplicación y máquina dependen de funciones que no son mezcladas en un solo módulo: (solo implementación) Toda referencia a máquina depende de las funciones dentro de un módulo reduce su generalidad. Un ejemplo podría ser el reloj del sistema de referencia para efectos calendario. Esta medida esta basada en el número de módulos que violan este concepto en diseño e implementación.
- (3) Procesamiento de datos de volumen no limitado: Un módulo que ha sido diseñado y codificado para no aceptar más de 100 entradas de datos para procesamiento no es ciertamente de naturaleza general como un módulo que aceptará cualquier volumen de entrada. Esta medida esta basada en el número de módulos que están diseñados o implementados para ser datos de volumen limitado.
- (4) Procesamiento de datos de valor no limitado: Un elemento previamente identificado, MÉTRICA 258. La tolerancia de error

negocia con el control de entrada para razonabilidad. Esta capacidad es necesaria para impedir el suministro de datos a una función para la que no está definida o su grado de precisión no es aceptable, etc., es necesaria la capacidad de un punto de vista de la tolerancia de error. Desde un punto de vista general, cuanto más pequeño es el subconjunto de todas las posibles entradas a una función que puede aplicarse es menos general. Por lo tanto, esta medida esta basada en el número de módulos de datos que son de valor limitado. Esto puede determinarse en el diseño y la implementación.

- (5) Todas las constantes deben definirse de una vez: Este elemento, en efecto, define una constante como un valor paramétricos. En un solo lugar en el módulo o la base de datos puede ser modificado para dar cabida a una aplicación distinta de la función de dicho módulo, por ejemplo, para calcular una relación matemática en un mayor grado de precisión o para representar la constante de gravitación de un planeta diferente a la tierra, etc. Por lo tanto, siguiendo esta regla, el esfuerzo necesario para aplicar el módulo en un entorno diferente es menor. La medida esta basada en el número de módulos que violan este concepto durante el diseño e implementación

Fórmula: A nivel de sistema:

$$X=A/B,$$

A= puntaje total de los elementos aplicables,

B (conteo): Número de elementos aplicables.

ELEMENTOS:

- (1), (2) y (5): $Y=(1-C/D)$,
C: Número de módulos que violan la regla,
D: Número total de módulos,
- (3) y (4): $Y=(1-C/D)$,
C (Conteo): Número de módulos limitados,
D (Conteo): Número total de módulos

Atributo al que ayuda a calcular: Generalidad

Granularidad: Sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica en las disciplinas de diseño e implementación a nivel de sistema

MÉTRICA 126

Nombre en español: Medida de Abstracción funcional

Nombre en inglés: Measure of functional abstraction (MFA)

Descripción: Esta métrica es el cociente del número de métodos heredados por una clase, al número total de métodos accesibles por métodos miembro de la clase. *Fórmula:*

$$MFA = A/B$$

A: Número de métodos heredados de la clase

B: Número de métodos totales, accesibles por métodos miembro de la clase

Atributo al que ayuda a calcular: Herencia

Granularidad: Clase

Rango: entre 0 y 1

Referencias: Bansiya [31], JETTER [3], OLMEDILLA [53], Garzas [57]

Espectro: Gris

Otros: Puede ser remplazada con (JETTER [3])

$$X = 1 - (\text{Número de métodos invalidados} / \text{número de métodos})$$

MÉTRICA 127

Nombre en español: Porción de métodos heredados FHM

Nombre en inglés: Method inheritance Factor - MIF

Descripción: La métrica es la proporción entre la suma de todos los métodos heredados en todas las clases y el número total de métodos (localmente definidos más los heredados) en todas las clases. Es un indicador del nivel de Reutilización. También es propuesto como ayuda para evaluar la cantidad de recursos necesarios a la hora de probar.

Fórmula:

$$MIF = \frac{\{\sum \text{ para } i=1 \text{ hasta TC de } [Md (Ci)]\}}{\{\sum \text{ para } i=1 \text{ hasta TC de } [Ma (Ci)]\}}$$

Donde:

$Ma (Ci) = Md (Ci) + Mi (Ci)$,

$Ma (Ci)$: Número de métodos disponibles,

$Md (Ci)$: Número de métodos definidos,

$Mi (Ci)$: es el número de métodos heredados,

TC: número total de clases

Atributo al que ayuda a calcular: Herencia

Influencia sobre otros atributos: Reutilización, Analizabilidad

Granularidad: Clase, Programa, Sistema

Referencias: VAZQUEZ [61], Abreu y Melo [79], PRESSMAN [1], Champeaux [80], Marin [28]

Espectro: Claro

Otros: Aplica en la disciplina de implementación

MÉTRICA 128

Nombre en español: Proporción de atributos heredados - FHA

Nombre en inglés: Attribute inheritance Factor - AIF

Descripción: La MÉTRICA es la proporción entre la suma de todos los atributos heredados y el número total de atributos. Es un indicador la capacidad de Reutilización en un sistema, al igual que MIF

Fórmula:

$$AIF = \frac{\{\sum \text{ para } i=1 \text{ hasta TC de } [Ad (Ci)]\}}{\{\sum \text{ para } i=1 \text{ hasta TC de } [At (Ci)]\}}$$

$\{\sum \text{ para } i=1 \text{ hasta TC de } [Aa (Ci)]\}$,

Donde:

$Aa (Ci) = Ad (Ci) + Ai (Ci)$,

$Aa (Ci)$: Número de atributos disponibles,

$Md (Ci)$: Número de atributos definidos,

$Ai (Ci)$: es el número de atributos heredados,

TC: número total de clases

Atributo al que ayuda a calcular: Herencia

Influencia sobre otros atributos: Reutilización, Analizabilidad

Granularidad: Clase, Programa, Sistema

Referencias: VAZQUEZ [61, Abreu y Melo [79], PRESSMAN [1], Champeaux [80], Marin [28]

Espectro: Claro

Otros: Aplica en las disciplinas de diseño e implementación

MÉTRICA 129

Nombre en español: Índice de especialización por clase

Nombre en inglés: Specialisation index per class SIX

Descripción: Muestra en qué medida las subclases redefine el comportamiento de sus superclases. Esta fórmula pondera más las redefiniciones que ocurren en niveles más profundos del árbol de Herencia, ya que, cuanto más especializada es una clase, menos probabilidad existe de que su comportamiento sea reemplazado. Cuando son utilizados frameworks (clases de bibliotecas especializadas), algunos métodos deben ser redefinidos: estos métodos no deben tenerse en cuenta al calcular esta métrica. SIX es propuesto como medida de la calidad en la Herencia. SIX puede indicar cuándo hay demasiados métodos redefinidos, de tal forma que las abstracciones pueden no ser apropiadas y sea necesario reemplazar su comportamiento. Generalmente, una subclase debería extender el comportamiento de la superclase con nuevos métodos más que reemplazar o borrar comportamiento a través de redefiniciones. Lorenz y Kidd [87] sugieren un valor del 15% para ayudar a identificar superclases que no tienen mucho en común con sus subclases. Cuanto más profundizamos en la jerarquía, más especializada ha de ser la subclase.

Fórmula:

$$SIX = (A * B) / C,$$

Donde:

A: Número de métodos redefinidos

B: Anidamiento en la jerarquía,

C: Número total de métodos

Atributo al que ayuda a calcular: Herencia

Granularidad: Clase

Referencias: VAZQUEZ [61], Lorenz y Kidd [87], EL-WAKIL [71], Genero [90]

Espectro: Gris

Otros: Aplica en las disciplinas de diseño e implementación

MÉTRICA 130

Nombre en español: Árbol de Profundidad de Herencia- APH

Nombre en inglés: Depth of inheritance tree -DIT

Descripción: la MÉTRICA mide el máximo nivel en la jerarquía de Herencia. Es la cuenta directa de los niveles en la jerarquía de Herencia. En el nivel cero de la jerarquía de encuentra la clase raíz. Chidamber y Kemerer [67] proponen DIT como medida de la Complejidad de una clase, la Complejidad del diseño y la Reutilización potencial, lo que es debido a que cuanto más profunda esta una clase en la jerarquía, mayor es la probabilidad de heredar más métodos. Es una medida de cuántos ancestros pueden afectar a esta clase. Es considerado que más de 6 niveles es un abuso de la Herencia.

Fórmula:

DIT=Máximo nivel de la jerarquía, rango

Atributo al que ayuda a calcular: Herencia, Jerarquía

Influencia sobre otros atributos: Reutilización, Comprensibilidad, Testeabilidad, Analizabilidad, Cambiabilidad, Estabilidad, Funcionalidad

Granularidad: Clase

Rango: desde 0 hasta N, donde N es un entero positivo

Referencias: VAZQUEZ [61], Chidamber y Kemerer [67], GONZALEZ [68], PRESSMAN [1], Lindell [86], Pritchett [69], MANSO [85], Basili [70], El-Wakil [71], Li [72], Ojha [78], Lindroos [73], ANDERSSON[74], Bar [75], Kan [24], LAIRD [76], Fenton [77], Garzas [57], Marin [28], JETTER [3], OLMEDILLA [53]

Espectro: Claro

Otros: Aplica en la disciplina de diseño

MÉTRICA 131

Nombre en español: Número de métodos heredados

Nombre en inglés: Number of Methods Inherited - NMI

Descripción: Esta métrica mide el número de métodos heredados por una subclase. No es mencionado si la Herencia es pública o privada

Fórmula:

NMI = Número de métodos heredados en la clase

Atributo al que ayuda a calcular: Herencia

Granularidad: Clase

Rango: desde 0 hasta N, donde N es un entero positivo

Referencias: EL-WAKIL [71], Genero [90]

Espectro: Gris

Otros: Aplica en disciplina de diseño

MÉTRICA 132

Nombre en español: Número de método remplazados

Nombre en inglés: Number of Methods Overriden - NMO

Descripción: Un gran número de métodos a remplazar indica un problema de diseño, lo que indica que los métodos fueron anulados en el último momento en el diseño. La subclase debería realmente ser una especialización de su súper clase, dando lugar a nuevos nombres únicos para sus métodos

Fórmula:

NMO = Número de métodos a remplazar en la clase

Atributo al que ayuda a calcular: Herencia

Granularidad: Clase

Rango: desde 0 hasta N, donde N es un entero positivo

Referencias: EL-WAKIL [71], Genero [90]

Espectro: Gris

Otros: Aplica en disciplina de diseño

MÉTRICA 133

Nombre en español: Número de métodos añadidos

Nombre en inglés: Number of methods added - NMA

Descripción: Este indicador contabiliza el número total de los métodos definidos en una subclase

Fórmula:

NMO = Número de métodos añadidos en la clase

Atributo al que ayuda a calcular: Herencia

Granularidad: Clase

Rango: desde 0 hasta N, donde N es un entero positivo

Referencias: Genero [90]

Espectro: Gris

Otros: Es posible que sea la misma métrica de Número de métodos nuevos (métrica 134)

MÉTRICA 134

Nombre en español: Número de nuevos métodos

Nombre en inglés: Number of New Methods - NNA

Descripción: La expectativa normal de una subclase es que además de ser especializada añada los métodos al objeto de la superclase. Un método es añadido en una subclase, si no existe alguno del mismo nombre en cualquiera de sus superclases.

Fórmula:

NMO = Número de métodos nuevos en la clase

Atributo al que ayuda a calcular: Herencia

Granularidad: Clase

Rango: desde 0 hasta N, donde N es un entero positivo

Referencias: EL-WAKIL [71]

Espectro: Gris

Otros: Es posible que sea la misma métrica de Número de métodos nuevos (métrica 133)

MÉTRICA 135

Nombre en español: Número de descendientes

Nombre en inglés: Number of descendantes - NOD

Descripción: Refiere al número de descendientes de una clase, es decir todas aquellas clases que tiene relación de Herencia (descendientes) con la clase que está trabajando.

Fórmula:

NOD: cuenta el número de descendientes (subclases directas e indirectas) de una clase

Atributo al que ayuda a calcular: Herencia

Granularidad: clase

Referencias: Bar [75]

Espectro: Gris

Otros: Es la misma métrica de Number of children - NOC (MÉTRICA 17)

MÉTRICA 136

Nombre en español: Número promedio de clases derivadas

Nombre en inglés: Average Number of Derived Classes - ANDC

Descripción: el número promedio de las subclases directas de una clase. Todas las clases definidas en el sistema de medida (y sólo ellos son considerados). Las interfaces no son tomadas en cuenta. Si una clase no tiene las clases derivadas, la clase participa con un valor ANDC de 0.

Fórmula:

$$ANDC = \{ \sum \text{desde } i=0 \text{ hasta } n \text{ de } (A_i * k_i) \} / B$$
, donde,
A_i (Conteo): Número de clases con igual número de descendientes
k_i (conteo): Número de descendientes de A.
B (conteo): Número de clases en el modelo que están en una jerarquía y no son interfaces

Atributo al que ayuda a calcular: Herencia

Granularidad: sistema

Referencias: Lanza [88]

Espectro: Gris

MÉTRICA 137

Nombre en español: Promedio de altura de la jerarquía

Nombre en inglés: Average Hierarchy Height - AHH

Descripción: La métrica es calculada como la media de la altura del árbol de herencia (HIT) entre las clases definidas en la raíz del sistema. AHH es el promedio de la longitud máxima del camino de una raíz más profunda a sus subclases. Una clase es una raíz, si no es derivada de otra clase pertenecientes. Las interfaces no son tomadas en cuenta. Clases independientes (es decir, las clases con ninguna clase de base en el sistema y no hay descendientes) son consideradas como aquellas de raíz con un valor HIT de 0. El número nos indica la profundidad de las jerarquías de clase. Los números bajos indican un piso estructura jerarquía de clases.

Fórmula:

$$AHH = \frac{\{\sum_{\text{desde } i=1 \text{ Hasta } n \text{ de } (HIT (A_i))\}}{B,}$$

Donde:

A_i: clase con descendientes,

B (conteo): Número de clases en el modelo que están en una jerarquía y no son interfaces

Atributo al que ayuda a calcular: Herencia

Granularidad:

Referencias: Lanza [88]

Espectro: Gris

Otros: HIT puede estar relacionada a la métrica DIT (Metica 130)

MÉTRICA 138

Nombre en español: Independencia de dispositivo 1

Nombre en inglés: Device-independence 1

Descripción: ¿Son los cómputos independientes del tamaño de palabra del computador para alcanzar el esquema de precisión o almacenamiento requerido?

Atributo al que ayuda a calcular: Independencia de dispositivo

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 139

Nombre en español: Independencia de dispositivo 2

Nombre en inglés: Device-independence 2

Descripción: Si el código de lenguaje de máquina es usado (e.g. por Eficiencia), ¿es un lenguaje de alto nivel actualizado con versión mantenida y disponible?

Atributo al que ayuda a calcular: Independencia de dispositivo

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 140

Nombre en español: Independencia de dispositivo 3

Nombre en inglés: Device-independence 3

Descripción: ¿Es necesaria la utilización de máquinas aisladas con características especiales?

Atributo al que ayuda a calcular: Independencia de dispositivo

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 141

Nombre en español: Independencia de dispositivo 4

Nombre en inglés: Device-independence 4

Descripción: ¿Han sido señalizados y comentadas las sentencias dependientes de la máquina (e.g. aquellos cómputos que dependen de la capacidad del hardware de la máquina para tratar algunas palabras, archivos, padres de bits seleccionados, o otros aquellos que emplean características extendidas del lenguaje origen)?

Atributo al que ayuda a calcular: Independencia de dispositivo

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 142

Nombre en español: Independencia de dispositivo 5

Nombre en inglés: Device-independence 5

Descripción: ¿Es el lenguaje de programación soportado por más de un fabricante de hardware de computadora?

Atributo al que ayuda a calcular: Independencia de dispositivo

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 143

Nombre en español: Independencia de dispositivo 6

Nombre en inglés: Device-independence 6

Descripción: ¿Está el programa estructurado de una manera que permita la operación por fases sobre un computador pequeño?

Atributo al que ayuda a calcular: Independencia de dispositivo

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 144

Nombre en español: Medida de independencia de maquina

Nombre en inglés: Machine independence measure

Descripción: La métrica es la suma de los puntajes de los siguientes elementos aplicables dividido por el número de elementos aplicables:

1. Lenguaje de programación disponible utilizado en otras máquinas: Esta es una medida binaria que identifica si el lenguaje de programación utilizado está disponible (1) en otras máquinas o no (0). Esto significa la misma versión y dialecto del lenguaje.
2. Libre de referencias de entrada / salida: Referencias de entrada y salida obligan a un módulo de la actual configuración de máquina. Por lo tanto los pocos módulos dentro de un sistema que contienen referencias de entrada y salida, más localizado el problema es convertido cuando es considerada la conversión. Esta medida representa ese hecho y esta

basada en el número de módulos dentro del sistema que contienen referencias 1 / 0. esto Aplica durante el diseño e implementación.

3. Código es independiente del tamaño de la palabra y carácter: (implementación) Instrucciones u operaciones que dependen del tamaño de la palabra o carácter de la máquina deben evitar parametrizarse, para facilitar su uso en otra máquina. Esta medida Aplica al código fuente durante la implementación esta basada en el número de módulos que contengan violaciones con el concepto de independencia de tamaño de palabra y de carácter.
4. Representación de datos de máquina independiente: (implementación). Las convenciones de nomenclatura (longitud) utilizadas son estándar o compatibles con otras máquinas. Esta medida esta basada en el número de módulos que contienen variables que no ajustan a las representaciones de datos estándar

Fórmula: A nivel de sistema:

$$X=A/B,$$

A= puntaje total de los elementos aplicables,
B (conteo): Número de elementos aplicables.

ELEMENTOS:

- (1): seguir definición;
- (2): $Y=(1-C/D)$,
C: Número de módulos con referencias de E/S,
D: Número total de módulos;
- (3) y (4): $Y=(1-C/D)$,
C: Número de módulos que violan la regla,
D: Número total de módulos;

Atributo al que ayuda a calcular: Independencia de dispositivo

Granularidad: Sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica a disciplinas de diseño e implementación a nivel de sistema

MÉTRICA 145

Nombre en español: Medida de independencia de sistema Software

Nombre en inglés: Software system independence measure

Descripción: La métrica es la suma de los puntajes de los siguientes elementos aplicables dividido por el número de elementos aplicables:

1. Dependencia sobre programas de utilidades de sistema software: Los programas de utilidad que más son utilizados dentro de un sistema más dependiente del sistema es en ese sistema software de entorno. Un TIPO de utilidad en un sistema operativo no es probable que sea exactamente similar a la utilidad de un TIPO en otra. Esta medida esta

basada en el número de programas usados que son programas de utilidad, dividido por el número total de programas en el sistema. Esto ha de aplicarse durante el diseño e implementación

2. Dependencia en rutinas de librería en el sistema software: Por motivos similares a (1) una función de integración dada por un sistema operativo puede no ser exactamente la misma función de integración dada por otro sistema. Por lo tanto a más rutinas de librería usadas más dependiente es el sistema en su actual entorno de software de sistema. Esta medida, aplicada en el diseño y la implementación, esta basada en el número de rutinas de librería utilizadas dividido por el número total de módulos en el sistema
3. Subconjunto común, estándar del lenguaje utilizado: El uso de construcciones no estándar de un lenguaje que puede estar disponible desde determinados compiladores causa problemas de conversión cuando el programa es trasladado a un nuevo entorno de software de sistema. Esta medida representa esa situación. Esta basada en el número de módulos que están codificados en un subconjunto no estándar del lenguaje. el subconjunto de estándares del lenguaje son establecidos durante el diseño y sigue durante la implementación.
4. Libre de Referencias de Sistema Operativo: Esta medida esta basada en el número de módulos que contiene las llamadas al sistema operativo. mientras (1) y (2) determinan el número de programas de tipo-apoyo y rutinas que pueden ser recodificadas si un cambio en el entorno del sistema software llevó a cabo, esta medida identifica el porcentaje de aplicación orientada a los módulos que probablemente han de ser cambiados. La métrica debe aplicarse durante el diseño y implementación

Fórmula: A nivel de sistema:

$$X=A/B,$$

A= puntaje total de los elementos aplicables,
B (conteo): Número de elementos aplicables.

ELEMENTOS:

- (1): $Y=(1 - C/D)$,
C: Número de programas = programa de utilidad,
D: Número total de programas
- (2): $Y=(1 - C/D)$,
C: Número de rutinas de librería usadas,
D: Número total de módulos;
- (3): $Y=(1 - C/D)$,
C: Número de módulos que violan la regla,
D: Número total de módulos;

- (4): $Y=(1 - C/D)$,
 C: Número de módulos con referencias de SO,
 D: Número total de módulos

Atributo al que ayuda a calcular: independencia de sistema Software

Granularidad: Sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica a disciplinas de diseño e implementación a nivel de sistema

MÉTRICA 146

Nombre en español: Fácil de instalar

Nombre en inglés: Ease of installation

Descripción: ¿Puede el usuario o mantenedor instalar el software fácilmente para un entorno operacional? Observe el comportamiento de usuario o mantenedor cuando está tratando de instalar el software para un entorno operacional

Fórmula:

$$X=A/B,$$

A (Conteo): Número de casos los cuales un usuario tiene éxito en cambiar la operación de instalación para su conveniencia,

B (Conteo): número total de casos en los cuales un usuario intenta cambiar la operación e instalación para su conveniencia,

Atributo al que ayuda a calcular: Instalabilidad

Rango: $0 \leq X \leq 1$, entre más cerca a 1 es mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 147

Nombre en español: Facilidad de re instalación - Externo

Nombre en inglés: Ease of setup re-try

Descripción: ¿Puede el usuario o mantenedor intentar de nuevo fácilmente configurar la instalación del software? Observe el comportamiento del usuario o mantenedor cuando esta tratando de configurar de nuevo la instalación del software

Fórmula:

$$X=1 - A/B,$$

A (Conteo): Número de casos en los cuales el usuario falla en configurar de nuevo durante la operación de configuración,

B (Conteo): número total de casos en los cuales el usuario intenta configurar de nuevo durante la operación de configuración

Atributo al que ayuda a calcular: Instalabilidad

Rango: $0 \leq X \leq 1$, entre más cerca a 1 es mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 148

Nombre en español: Facilidad de re instalación - interno

Nombre en inglés: Ease of setup re-try

Descripción: ¿Qué tan fácil es repetir la operación de instalación? Cuenta el número de operaciones de re instalaciones implementadas y lo compara al número de operaciones de re instalaciones requeridas

Fórmula:

$$X=A/B,$$

A (Conteo): Número operaciones implementadas de re instalación, confirmado en revisión,

B (Conteo): número de operaciones de instalación requeridas
 $0 \leq X < 1$, entre más cerca a 1 mas fácil

Atributo al que ayuda a calcular: Instalabilidad

Rango: $0 \leq X \leq 1$, entre más cerca a 1 mas fácil

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 149

Nombre en español: Esfuerzo de instalación

Nombre en inglés: Installation effort

Descripción: ¿Qué nivel de esfuerzo es requerido para la instalación? Cuenta el número de pasos automatizados de instalación implementados y lo compara con el número de pasos de instalación recomendados

Fórmula:

$$X=A/B,$$

A (Conteo): Número pasos de instalación automatizados, confirmado en revisión,

B (Conteo): número de pasos de instalación requeridos $0 \leq X < 1$, entre más cerca a 1 mejor

Atributo al que ayuda a calcular: Instalabilidad

Rango: $0 \leq X \leq 1$, entre más cerca a 1 es mejor.

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 150

Nombre en español: Flexibilidad de instalación

Nombre en inglés: Instalación flexibility

Descripción: ¿Qué tan flexible y personalizable es la capacidad de instalación? Cuenta el número de operaciones de instalación personalizable implementadas tal como en la especificación y lo compara al número de operaciones de instalación con requerimientos de capacidad de personalización

Fórmula:

$$X=A/B,$$

A (Conteo): Número de operaciones de instalación personalizable implementadas tal como en la especificación, confirmado en revisión,
B (Conteo): número de operaciones de instalación personalizables requeridas

Atributo al que ayuda a calcular: Instalabilidad

Rango: $0 \leq X < 1$, entre más cerca a 1 más flexible

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 151

Nombre en español: Medida de prueba de modulo

Nombre en inglés: Module testing measure (Module testing support) (By module)

Descripción: la métrica de nivel de sistema es un promedio de todas las medidas de los módulos. La medida del modulo es el puntaje promedio de los siguientes elementos:

1. Cobertura de ruta: Planes para probar los distintos caminos dentro de un módulo deben hacerse durante el diseño y los casos de prueba realmente desarrollados durante Implementación. esta medida identifica el número de rutas previstas para el ensayo, dividido por el número total de caminos.
2. Límite de parámetros de entrada probados: otro aspecto de las pruebas de modulo implica probar los rangos de entrada para el módulo. Esto es realizado mediante el ejercicio del módulo en los distintos valores límite de los parámetros de entrada. Los planes para hacer esto deben especificarse en el diseño y codificar durante la implementación. La medida es el número de parámetros de límite probados dividido por el número total de parámetros.

Fórmula: A nivel de sistema:

$$X = A/B,$$

A= puntaje total de los elementos aplicables,

B (conteo): Número de elementos aplicables.

$$M = \sum Q / Z,$$

Q: Medida de prueba del modulo para cada modulo,

Z: Número total de módulos;

ELEMENTOS:

- (1): $Y = (1 - C/D)$,

C: Número de rutas a ser probadas,

D: Número total de rutas,

- (2): $Y = (1 - C/D)$,

C: Número de parámetros a ser probados el limite,

D: Número total de parámetros

Atributo al que ayuda a calcular: Instrumentación

Granularidad: Modulo, Sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica a disciplinas de diseño, implementación, primero a nivel de modulo y después a nivel de sistema

MÉTRICA 152

Nombre en español: Medida de prueba de integración

Nombre en inglés: Integration testing measure (Integration testing support)

Descripción: la métrica de nivel de sistema es un promedio de todas las medidas de los módulos. La medida del modulo es el puntaje promedio de lo siguientes elementos:

1. Módulos de interface probados: Uno de los aspectos de pruebas de integración es la prueba de todos las interfaces módulo a módulo. Planes para realizar esta prueba son preparados durante el diseño y las pruebas son desarrolladas durante la implementación. La medida esta basada en el número de interfaces a ser probadas dividida por el número total de ellas.
2. Requerimientos de desempeño (Cronometro y almacenamiento) cubiertos: El segundo aspecto de la prueba de integración implica la comprobación del cumplimiento en el módulo y el subsistema con los requisitos de rendimiento. Esta prueba está previsto durante el diseño y las pruebas son desarrolladas durante la implementación. La medida es el número de requisitos de desempeño a ser probados dividida por el número total de requisitos de desempeño.

Fórmula: A nivel de sistema:

$$X=A/B,$$

A= puntaje total de los elementos aplicables,

B (conteo): Número de elementos aplicables.

ELEMENTOS:

- (1): $Y=(1 - C/D),$

C: Número a ser probadas,

D: Número total de interfaces,

- (2): $Y=(1 - C/D),$

C: Número de requerimientos a ser probados,

D: Número total requerimientos de rendimiento

Atributo al que ayuda a calcular: Instrumentación

Granularidad: Sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica en disciplina implementación a nivel de sistema

MÉTRICA 153

Nombre en español: Medida de prueba de sistema

Nombre en inglés: System testing measure (System testing support)

Descripción: La métrica es el puntaje promedio de los siguientes elementos:

1. Cobertura de modulo (Para todos los escenarios probados): Uno de los aspectos de sistema de pruebas que pueden medirse en la fase de diseño es el equivalente a la cobertura del camino a nivel de módulo. Para todos los escenarios de prueba del sistema previsto, el porcentaje de todos los módulos sobre los que es ejercida es importante.
2. Identificación de entradas y salidas de prueba en resumen: Los resultados de las pruebas y la manera en que estos resultados son muestreados son muy importantes para la eficacia de las pruebas. Esto es especialmente cierto en el sistema de ensayo debido a la posible gran volumen de datos de entrada y salida. Esta medida identifica simplemente si existe la capacidad para mostrar las entradas y salidas de prueba en un resumen de moda. La medida puede aplicarse a los planes y especificaciones de la fase de diseño y el desarrollo de esta capacidad durante la implementación.

Fórmula: A nivel de sistema:

$$X=A/B,$$

A= puntaje total de los elementos aplicables, B (conteo): Número de elementos aplicables.

ELEMENTOS:

- (1): $Y=(1 - C/D)$,
C: Número de módulos a ser ejecutados,
D: Número total de módulos
- (2): Seguir la definición

Atributo al que ayuda a calcular: Instrumentación

Granularidad: Sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica a disciplinas de diseño, implementación, a nivel de sistema

MÉTRICA 154

Nombre en español: Medida de Estabilidad

Nombre en inglés: Stability measure (Degree of independence)

Descripción: Ésta medida esta basada en la categorización de Myers [91] de los módulos por su resistencia y acoplamiento. Módulo de resistencia es una medida de la Cohesión o la relación de los elementos dentro de un módulo. Módulo de acoplamiento es una medida de la relación entre los módulos. La métrica combina estas dos medidas para calcular el número esperado de los

módulos que requieren codificación si los cambios a cualquier módulo son divididos por el número total de módulos.

Fórmula: A nivel de sistema:

$$X=A/B,$$

A (Conteo): Número esperado de módulos cambiados,

B (Conteo): número de módulos.

Atributo al que ayuda a calcular: Modularidad

Granularidad: Sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica a disciplina de diseño a nivel de sistema

MÉTRICA 155

Nombre en español: Medida de implementación modular

Nombre en inglés: Modular implementation measure

Descripción: La métrica es la suma de los puntajes de los siguientes elementos aplicables divididos por el número de elementos aplicables:

1. Estructura jerárquica: la medida refiere a la estructura de diseño arriba hacia abajo mencionado en métrica 188. la estructura jerárquica obtenida ejemplificara el flujo de control entre un modulo predecesor y su modulo inmediatamente sucesor. esta medida esta basada en el número de violaciones a esta regla.
2. Todos los módulos que no excedan de un tamaño de módulo estándar (100): (solo fase implementación) El módulo de tamaño estándar de 100 declaraciones de procedimiento puede variar. 100 fue elegido porque fue mencionado en la literatura a frecuentemente. Esta medida esta basada en el número de módulos que excedan el tamaño estándar establecido.
3. Todos los módulos representan una función: El concepto de modularidad esta basada en cada una de las funciones es implementado en un único módulo. Esta medida esta basada en el número de módulos que representan más de una función. Esto puede determinarse tanto en el diseño e implementación.
4. Control de los parámetros definidos por el módulo de llamadas: Los próximos cuatro elementos de más adelante son elaborados sobre el control y la interacción entre los módulos contemplados en (1). El llamamiento módulo define los parámetros de control, los datos de entrada requeridos, y la salida de los datos requeridos. el control también deben ser devuelto a la llamada de módulo. Esta medida y el próximas tres esta basada n en el número de violaciones a estas reglas. Ellos pueden ser medidos en el diseño implementación

5. Datos de entrada controlados por el modulo llamada: Ver (4).
6. Datos de salida suministrados al modulo de llamada: Ver (4),
7. Control retornado al módulo de llamada: Ver (4),
8. Los módulos no comparten almacenamiento temporal: Esta es una medida binaria, 1 si los módulos no comparten almacenamiento temporal y 0 si lo hacen. Hace hincapié en la pérdida de independencia si el módulo de almacenamiento temporal es compartida entre los módulos.

Fórmula: A nivel de sistema:

$$X=A/B;$$

A: Puntaje total de elementos aplicables,
 B: Número de elementos aplicables;
 D: Número de módulos.

ELEMENTOS:

- (8): Seguir la definición
- (1): $M=(1 - Q/Z)$,
 Q: Número de violaciones a la jerarquía,
 Z: Número total de módulos,
- (2): $M=(1 - Q/Z)$,
 Q: Número de módulos mayor que 100,
 Z: Número total de módulos,
- (3),(4),(5),(6) y (7): $M=(1 - Q/Z)$,
 Q: Número de módulos que violan la regla,
 Z: Número total de módulos

Atributo al que ayuda a calcular: Modularidad

Granularidad: Sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica a disciplinas de diseño, implementación, a nivel de sistema

MÉTRICA 156

Nombre en español: Consistencia operacional en uso

Nombre en inglés: Operational consistency in use

Descripción: ¿Cuan consistentes son los componentes de la interface de usuario? Observa el comportamiento del usuario y pregunta su opinión.

Fórmula:

a) $X=1 - A/B,$

b) $Y=N/UOT,$

A (Conteo): Número de mensajes o funciones las cuales el usuario encuentra inconsistencias inaceptables con la expectativa de usuario,

B (Conteo): Número de mensajes o funciones,

N (Conteo): número de operaciones en las cuales el usuario encuentra inconsistencias inaceptables con la expectativa de usuario

UOT (Tiempo): tiempo de operación del usuario (durante el periodo de observación). $0 \leq X \leq 1$, entre más cerca a 1 mejor. $0 \leq Y$, entre más pequeño y cercano a 0.0 mejor

Atributo al que ayuda a calcular: Operatividad

Rango: $0 \leq X \leq 1$, entre más cerca a 1 mejor. $0 \leq Y$, entre más pequeño y cercano a 0.0 mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 157

Nombre en español: Corrección del error

Nombre en inglés: Error correction

Descripción: ¿Puede el usuario corregir fácilmente errores sobre tareas? Test de conducta de usuarios y observación de conducta de usuarios.

Fórmula:

$$T = T_c - T_s,$$

T_c (Tiempo): tiempo de corrección completada de errores de tipo especificado de tareas realizadas,

T_s (Tiempo): Tiempo de corrección puesta en marcha de errores de tipo especificado de tarea realizada.

Atributo al que ayuda a calcular: Operatividad

Rango: $0 < T$, entre más corto mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 158

Nombre en español: Corrección del error en uso

Nombre en inglés: Error correction in use

Descripción: ¿Puede el usuario recuperar fácilmente sus errores reintentar tareas? Observación de conducta del usuario que está operando el software. ¿Puede el usuario recuperar fácilmente sus entradas? Observación de conducta del usuario que está operando el software.

Fórmula:

a) $X = 1 - A/UOT$,

b) $X = A/B$,

a) A (Conteo): Número de veces que el usuario tiene éxito al cancelar sus errores de operación,

UOT (Tiempo): tiempo de operación del usuario durante el periodo de observación,

b) A (Conteo): Número de pantallas o formas donde los datos de entrada son modificados o cambiados satisfactoriamente antes de ser elaborado,

B (Conteo): Número de pantallas o formas donde el usuario ensaya modificar o cambiar los datos de entrada durante la observación de usuario en tiempo de operación.

Atributo al que ayuda a calcular: operatividad

Rango: a) $0 \leq X$, entre más alto es mejor, b) $0 \leq X \leq 1$, entre más cerca a 1 mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 159

Nombre en español: Disponibilidad del valor por defecto en uso

Nombre en inglés: Default value availability in use

Descripción: ¿Puede el usuario fácilmente seleccionar valores de parámetros para su operación conveniente? Observación del comportamiento del usuario quien está operando el software. Cuenta cuantas veces el usuario intenta establecer o seleccionar valores de parámetro y falla, (porque el usuario no puede usar valores por defecto dados por el software)

Fórmula:

$$X = 1 - A/B,$$

A (Conteo): El número de veces que el usuario falla para establecer o seleccionar valores de parámetros en un corto periodo (porque el usuario no puede usar valores por defecto dados por el software),

B (Conteo): Número total de veces que el usuario intenta establecer o seleccionar valores de parámetro.

Atributo al que ayuda a calcular: Operatividad

Rango: $0 \leq X \leq 1$, entre más cerca a 1 mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 160

Nombre en español: Comprensibilidad de mensaje en uso

Nombre en inglés: Message understandability in use

Descripción: ¿Puede el usuario entender fácilmente mensajes desde el sistema software? ¿Hay algún mensaje que cause que el usuario este retrasado en entender antes de comenzar la siguiente acción? ¿Puede el usuario memorizar fácilmente mensajes importantes? Observación del comportamiento del usuario quien está operando el software

Fórmula:

$$X = A/UOT,$$

A (Conteo): Número de veces que el usuario pausa por un largo periodo o falla sucesivamente y repetidamente en la misma operación,

UOT (Tiempo): tiempo de operación del usuario (durante el periodo de observación),

Atributo al que ayuda a calcular: Operatividad

Rango: $0 \leq X$, entre más corto y cerca a 0.0 es mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 161

Nombre en español: Auto explicación de mensajes de error

Nombre en inglés: Self-explanatory error messages

Descripción: ¿En que proporción de condiciones de error el usuario propone la acción de recuperación correcta? Test de conducta de usuario y observación de conducta de usuario

Fórmula:

$$X=A/B,$$

A (Conteo): Número de condiciones de error para las cuales el usuario propone la acción de recuperación correcta

B (Conteo): Número de condiciones de error probadas.

Atributo al que ayuda a calcular: Operatividad

Rango: $0 \leq X \leq 1$, entre más cerca a 1 mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 162

Nombre en español: Recuperabilidad de error operacional en uso

Nombre en inglés: Operational error recoverability in use

Descripción: ¿Puede el usuario recuperar fácilmente su peor situación? Observación del comportamiento del usuario quien está operando el software

Fórmula:

$$X=1 - A/B,$$

A (Conteo): Número de situaciones recuperadas sin éxito (después de un error o cambio del usuario) en el cual el usuario no ha sido informado sobre el riesgo por el sistema,

B (Conteo): número de errores o cambios del usuario.

Atributo al que ayuda a calcular: operatividad

Rango: $0 \leq X \leq 1$, entre más cerca a 1 mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 163

Nombre en español: Tiempo entre operaciones de error humano en uso

Nombre en inglés: Time between human error operations in use

Descripción: ¿Puede el usuario operar el software suficientemente extenso sin error humano? Observación del comportamiento del usuario quien está operando el software

Fórmula:

$$X=T/N \text{ (al tiempo } t \text{ durante } [t-T,t]),$$

T (Tiempo): Periodo de tiempo de operación durante la observación (o la suma del tiempo de operación entre las operaciones de error humano del usuario),
N (Conteo): Número de ocurrencias de operación de errores humanos del usuario,

Atributo al que ayuda a calcular: Operatividad

Rango: $0 < X$, entre más alto mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 164

Nombre en español: Capacidad de deshacer

Nombre en inglés: Undoability (user error correction)

Descripción: ¿Cuan frecuentemente el usuario corrige satisfactoriamente los errores de entrada? ¿Cuán frecuentemente el usuario deshace correctamente los errores? Test de conducta de usuario y observación del comportamiento del usuario

Fórmula:

$$a) X = 1 - A/B,$$

$$b) Y = A/B,$$

a) A (Conteo): Número de errores de entrada los cuales el usuario corrige exitosamente,

B (Conteo): número de intentos de corregir los errores de entrada

b) A (Conteo): Número de condiciones de error las cuales el usuario corrige exitosamente,

B (Conteo): Número de condiciones de error probadas,

Atributo al que ayuda a calcular: Operatividad

Rango: $0 \leq X \leq 1$, entre más cerca a 1 mejor. $0 \leq Y \leq 1$, entre más cerca a 1 mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 165

Nombre en español: Personalización – externo

Nombre en inglés: Customisability

Descripción: ¿Puede el usuario personalizar fácilmente procedimientos de operación para su conveniencia? ¿Puede un usuario, quien instruye usuarios finales, fácilmente personalizar plantillas de procedimiento de operación para prevenir errores? ¿Qué proporción de funciones pueden ser personalizadas? Test de conducta de usuario y observación de comportamiento de usuario

Fórmula:

$$X = A/B,$$

A (Conteo): Número de funciones personalizadas exitosamente

B (Conteo): Número de intentos para personalizar.

Atributo al que ayuda a calcular: operatividad
Rango: $0 \leq X < 1$, entre más cerca a 1 es mejor
Referencias: ISO 9126 [15]:2
Espectro: Gris

MÉTRICA 166

Nombre en español: Operación de reducción de procedimiento
Nombre en inglés: Operación procedure reduction
Descripción: ¿Puede el usuario reducir fácilmente procedimiento de operación para su conveniencia? Cuenta toques de usuario para operaciones específicas y los compara entre antes y después de la operación de personalización
Fórmula:

$$X = 1 - A/B,$$

A (Conteo): número de procedimientos de operación reducidos después de la operación de personalización,
B (Conteo): Número de procedimientos de operación antes de la operación de personalización.

Atributo al que ayuda a calcular: Operatividad
Rango: $0 \leq X < 1$, entre más cerca a 1 es mejor
Referencias: ISO 9126 [15]:2
Espectro: Gris

MÉTRICA 167

Nombre en español: Accesibilidad física – externo
Nombre en inglés: Physical accessibility
Descripción: ¿Qué proporción de funciones pueden ser accedidas por usuarios con incapacidades físicas? Test de conducta de usuario y observación de conducta del usuario
Fórmula:

$$X = A/B,$$

A (Conteo): Número de funciones accedidas exitosamente
B (Conteo): Número de funciones.

Atributo al que ayuda a calcular: Operatividad
Rango: $0 \leq X < 1$, entre más cerca a 1 es mejor
Referencias: ISO 9126 [15]:2
Espectro: Gris

MÉTRICA 168

Nombre en español: Lista de chequeo de validación de entrada
Nombre en inglés: Input validity checking
Descripción: ¿Cuál es la proporción de entradas para proveer comprobación de datos validos? Cuenta el número de entradas, las cuales inspeccionan datos validos y compara con el número de entradas, las cuales pueden inspeccionar datos validos.

Fórmula:

$$X=A/B,$$

A (Conteo): Número de entradas que inspeccionan datos validos,

B (Conteo): Número de entradas que pueden inspeccionar datos validos.

Atributo al que ayuda a calcular: Operatividad

Rango: $0 \leq X < 1$, entre más cerca a 1 es mejor

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 169

Nombre en español: Operación de usuario, capacidad de cancelar

Nombre en inglés: User Operation cancellability

Descripción: ¿Cuál es la proporción de funciones que pueden ser canceladas antes de terminar? Cuenta el número de funciones implementadas, las cuales pueden ser canceladas por el usuario antes de terminar y lo compara con el número de funciones que requieren capacidad de pre-cancelación

Fórmula:

$$X=A/B,$$

A (Conteo): Número de funciones implementadas que pueden ser canceladas por el usuario,

B (Conteo): Número de funciones que requieren la capacidad de pre-cancelación.

Atributo al que ayuda a calcular: Operatividad

Rango: $0 \leq X < 1$, entre más cerca a 1 es mejor la habilidad de cancelación

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 170

Nombre en español: Operación de usuario, deshacer

Nombre en inglés: Customisability

Descripción: ¿Cuál es la proporción de de funciones que pueden ser personalizadas durante la operación? Cuenta el número de funciones implementadas, que pueden ser personalizadas por el usuario durante la operación y lo compara con el número de de funciones que requieren la capacidad de personalización

Fórmula:

$$X=A/B,$$

A (Conteo): Número de funciones que pueden ser personalizadas durante la operación,

B (Conteo): Número de funciones que requieren la capacidad de personalización.

Atributo al que ayuda a calcular: Operatividad

Rango: $0 \leq X < 1$, entre más cerca a 1 es mejor la habilidad de personalizar

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 171

Nombre en español: Personalización – interno

Nombre en inglés: Customisability

Descripción: ¿Cuál es la proporción de de funciones que pueden ser personalizadas durante la operación? Cuenta el número de funciones implementadas, que pueden ser personalizadas por el usuario durante la operación y lo compara con el número de de funciones que requieren la capacidad de personalización

Fórmula:

$$X=A/B,$$

A (Conteo): Número de funciones que pueden ser personalizadas durante la operación,

B (Conteo): Número de funciones que requieren la capacidad de personalización.

Atributo al que ayuda a calcular: Operatividad

Rango: $0 \leq X < 1$, entre más cerca a 1 es mejor la habilidad de personalizar

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 172

Nombre en español: Accesibilidad física – interno

Nombre en inglés: Physical accessibility

Descripción: ¿Cuál es la proporción de funciones que pueden ser personalizadas para acceso por el usuario con discapacidad física? Cuenta el número de funciones implementadas, que pueden ser personalizadas y lo compara con el número de funciones

Fórmula:

$$X=A/B,$$

A (Conteo): Número de funciones que pueden ser personalizadas,

B (Conteo): Número de funciones.

Atributo al que ayuda a calcular: Operatividad

Rango: $0 \leq X < 1$, entre más cerca a 1 es mejor la habilidad de acceso físico

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 173

Nombre en español: Estado de funcionamiento de la capacidad de vigilancia

Nombre en inglés: Operation status monitoring capability

Descripción: ¿Cuál es la proporción de funciones que tienen capacidad de monitorear el estatus de operación? Cuenta el número de funciones implementadas, para las cuales el estatus puede ser monitoreado y lo compara con el número de funciones que requieren la capacidad de monitoreo.

Fórmula:

$$X=A/B,$$

A (Conteo): Número de funciones que tiene capacidad de monitorear el estado,

B (Conteo): Número de funciones que requieren tener capacidad de monitoreo.

Atributo al que ayuda a calcular: Operatividad

Rango: $0 \leq X < 1$, entre más cerca a 1 es mejor la capacidad de monitoreo

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 174

Nombre en español: Consistencia operacional

Nombre en inglés: Operational consistency

Descripción: ¿Qué proporción de operaciones posee el mismo comportamiento en operaciones similares en otras partes del sistema? Cuenta el número de instancias de operaciones con comportamiento inconsistente y lo compara con el número total de operaciones

Fórmula:

$$X=1 - A/B,$$

A (Conteo): número de instancias de operaciones con comportamiento inconsistente,

B (Conteo): Número total de operaciones..

Atributo al que ayuda a calcular: Operatividad

Rango: $0 \leq X \leq 1$, entre más cerca a 1 más consistente

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 175

Nombre en español: Claridad de mensaje

Nombre en inglés: Message clarity

Descripción: ¿Cuál es la proporción de mensajes que son auto explicativos? Cuenta el número de mensajes implementados con explicaciones claras y lo compara con el número total de mensajes implementados

Fórmula:

$$X=A/B,$$

A (Conteo): Número de mensajes implementados con explicaciones claras, B (Conteo): Número de mensajes implementados.

Atributo al que ayuda a calcular: Operatividad

Rango: $0 \leq X < 1$, entre más cerca a 1 es más claro.

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 176

Nombre en español: Claridad de elemento de interface

Nombre en inglés: Interface element clarity

Descripción: ¿Cuál es la proporción de elementos de interface que son auto explicativo? Cuenta el número de elementos de interface que son auto explicativos y lo compara con el número total de elementos de interface

Fórmula:

$$X=A/B,$$

A (Conteo): Número de elementos de interface que son auto explicativas,

B (Conteo): Número total de elementos de interface

Atributo al que ayuda a calcular: Operatividad

Rango: $0 \leq X < 1$, entre más cerca a 1 es más claro.

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 177

Nombre en español: Recuperabilidad de error operacional

Nombre en inglés: Operational error recoverability

Descripción: ¿Qué proporción de funciones pueden tolerar errores del usuario? Cuenta el número de funciones implementadas con tolerancia al error del usuario y lo compara con el número total de total de funciones que requieren capacidad de tolerancia

Fórmula:

$$X=A/B,$$

A (Conteo): Número de funciones con tolerancia al error del usuario,

B (Conteo): Número total de funciones que requieren la capacidad de tolerancia.

Atributo al que ayuda a calcular: operatividad

Rango: $0 \leq X < 1$, entre más cerca a 1 es mas recuperable.

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 178

Nombre en español: Lista de chequeo de operatividad

Nombre en inglés: Operability checklist

Descripción: La métrica es la suma de los puntajes de los siguientes elementos aplicables, dividido por el número de elementos aplicables:

1. Todos los pasos de la operación descritos: esta es una medida binaria aplicada a todas las tres fases identificadas si las características operativas han sido descritas en la especificación de requerimientos, y si esa descripción ha sido transferidas in una descripción implementable de la operación (Usualmente en un manual de operación). la descripción de la operación cubrirá la secuencia normal de pasos y todos los pasos alternativos.

2. Todas las condiciones de error y respuestas apropiadamente descritas al operador: El requerimiento para esta capacidad debe aparecer en la especificación de requisitos, debe ser considerado durante el diseño, y codificado durante la implementación. las condiciones de error debe ser claramente identificadas por el sistema. las respuestas legales para todas las condiciones son documentadas y/o advertidas por el sistema. esta es una medida binaria para trazar la Evolución e implementación de estas capacidades.
3. Provisión para el operador de interrupción, obtención de estado, guardado, modificado, y continuo procesamiento: las capacidades provistas al operador deben ser consideradas durante la fase de requerimientos y después diseñar e implementar. ejemplos de las capacidades del operador incluye interrupción/resumen y puntos de comprobación. esta es una medida binaria para trazar la Evolución de estas capacidades.
4. Número de acciones del operador razonables: (solo implementación, requiere ejecución) el número de errores del operador pueden ser relacionados directamente al número de acciones requeridas durante un periodo de tiempo. esta medida esta basada en la cantidad de tiempo gastado requerido por las acciones del operador manual dividido por el total de tiempo requerido para el trabajo.
5. Encendido de trabajo y procesamientos arruinados: (solo implementación) la especificación de tareas involucradas en formar un trabajo y completarlo son descritas. esto es usualmente documentado durante la fase de implementación cuando la versión final del sistema es arreglado. eta es una medida binaria de la existencia de esa descripción.
6. Log Copias de las interacciones mantenidas: (diseño e implementación) esta es la capacidad que deben planearse en diseño y codificarse en implementación. Asistir en corregir errores operacionales, mejorar Eficiencia de operación, etc. esta medida identifica si es considerado en las fases diseño e implementación 1 o si no 0.
7. Mensajes de operador consistentes y respuestas estándar: (diseño e implementación) esta es una medida binaria aplicada durante el diseño y la implementación para asegurar que las interacciones entre el operador y el sistema son simples y consistentes. las respuestas del operador como YES, NO, GO, STOP, son concisas, simples, y pueden ser consistentemente usadas en todo el sistema. repuestas

formateadas larga, diferentemente no solo provee dificultad al operador sino también requiere rutinas de comprobación de error complejas.

Fórmula: A nivel de sistema:

$$X=A/B,$$

$A=\sum a_i$, (Para i de 1 a 4) a_i : puntajes de los elementos aplicables,
B (conteo): Número de elementos aplicables.

ELEMENTOS:

- (1): flujos normales y alternativos,
- (2)(3)(5)(6)(7): seguir la definición, (
- 4) $Y=(1 - C/D)$,
C: Tiempo para las acciones del operador,
D: Tiempo total para el trabajo

Atributo al que ayuda a calcular: Operatividad

Granularidad: sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica en disciplinas de análisis, diseño, implementación al nivel del sistema

MÉTRICA 179

Nombre en español: Número de métodos polimórficos -NOP

Nombre en inglés: Number of polymorphic methods (NPM)

Descripción: Esta métrica es una cuenta de los métodos que pueden exhibir comportamiento polimórfico. Tales métodos en c++ marcados como virtuales

Fórmula:

NPM= número de métodos de la clase que presentan comportamiento polimórfico.

Atributo al que ayuda a calcular: Polimorfismo

Granularidad: Método

Referencias: Bansiya [31], JETTER [3], OLMEDILLA [53], El-Wakil [71], Garzas [57]

Espectro: Gris

Otros: Puede ser remplazada con El número de Métodos invalidados (Number of Overridden Methods) Jetter [3]

MÉTRICA 180

Nombre en español: Proporción de Polimorfismo -FP

Nombre en inglés: Polymorphism Factor -POF

Descripción: Es la proporción entre el número real de posibles situaciones polimorfas para una clase C_i y el máximo número posible de situaciones polimorfas en C_i . Es decir, es el número de métodos heredados redefinidos dividido entre el máximo número de situaciones polimorfas distintas posibles. POF es una medida del Polimorfismo y una medida indirecta de la asociación dinámica en un sistema. El Polimorfismo es debido a la Herencia. En algunos

casos, sobrecargando métodos reduce la Complejidad y, por tanto, incrementa la Facilidad de Mantenimiento y la Facilidad de comprensión del sistema. En un sistema sin Herencia el valor de POF resulta indefinido, exhibiendo una discontinuidad

Fórmula:

$$POF = \frac{\{\sum \text{desde } l=1 \text{ hasta TC de } [Mo(Ci)]\}}{\{\sum \text{desde } l=1 \text{ hasta TC de } [Md(Ci)*DC(Ci)]\}}$$

Donde

Md (Ci)= Mn (Ci) + Mo (Ci),

DC (Ci): número de descendientes de Ci.

Mn (Ci): número de métodos nuevos.

Mo (Ci): número de métodos redefinidos.

TC: número total de clases.

El numerador representa el número de métodos heredados redefinidos. El denominador representa el máximo número de situaciones polimorfos distintas posibles para la clase Ci

Atributo al que ayuda a calcular: Polimorfismo

Influencia sobre otros atributos: Herencia

Granularidad: Clase

Referencias: VAZQUEZ [61]., Abreu y Melo [79], PRESSMAN [1], Champeaux [80]

Espectro: Claro

Otros: Aplica a disciplinas de diseño, implementación

MÉTRICA 181

Nombre en español: Capacidad de auditoría de acceso – externo

Nombre en inglés: Access auditability

Descripción: ¿Qué tan completo es el seguimiento de la auditoria concerniente a acceso de usuario para el sistema y datos? Evalúa la cantidad de accesos que el sistema graba en la base de datos del historial de acceso

Fórmula:

$$X=A/B,$$

A (Conteo): Número de "accesos de usuario al sistema y datos" grabado en la base de datos de historial de acceso

B (Conteo): Número de "accesos de usuario al sistema y datos" hechos durante la evaluación

Atributo al que ayuda a calcular: Seguridad

Rango: $0 \leq X \leq 1$, entre más cerca a 1 mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 182

Nombre en español: Capacidad de control de acceso - externo

Nombre en inglés: Access controllability

Descripción: ¿Qué tan controlable es el acceso al sistema? Cuenta el número de operaciones ilegales detectadas con comparación al número de operaciones ilegales como en las especificaciones

Fórmula:

$$X=A/B,$$

A (Conteo): Número de diferentes tipos detectados en operaciones ilegales

B (Conteo): Número de tipos de operaciones ilegales como en la especificación

Atributo al que ayuda a calcular: seguridad

Rango: $0 \leq X \leq 1$, entre más cerca a 1 mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 183

Nombre en español: Prevención de corrupción de datos – externo

Nombre en inglés: Data corruption prevention

Descripción: ¿Cuál es la frecuencia de eventos de corrupción de datos? Cuenta el número de ocurrencias de mayor y menor eventos de corrupción de datos

Fórmula:

a) $X=1-A/N$

b) $Y=1-B/N$

c) $Z=A/T$ or B/T

A (Conteo): Número de veces que ocurre un evento de corrupción de datos mayor

B (Conteo): Número de veces que ocurre un evento de corrupción de datos menor

N (Conteo): Número de casos de prueba ensayados para causar un evento de corrupción de datos.

Atributo al que ayuda a calcular: Seguridad

Rango: $0 \leq X \leq 1$, entre más cerca a 1 mejor, $0 \leq Y \leq 1$, entre más cerca a 1 mejor, $0 \leq Z$, entre más cerca a 0 mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

Otros: Corrupción de datos mayor: -Reproducción y recuperación imposible- hay afección secundaria a la distribución demasiado amplia- importancia de los datos en sí. Corrupción de datos menor: - Reproducción y recuperación posible- no hay afección secundaria a la distribución- importancia de los datos en sí.

MÉTRICA 184

Nombre en español: Capacidad de auditoría de acceso – interno

Nombre en inglés: Access auditability

Descripción: ¿Qué tan auditable es el acceso de login? Cuenta el número de tipos de acceso que son registrados correctamente como en las especificaciones y compara con el número de tipos de acceso que son requeridas a ser registrados en las especificaciones

Fórmula:

$$X=A/B,$$

A (Conteo): Número de tipos de accesos que son registrados como en las especificaciones

B (Conteo): Número de tipos de accesos de requeridos para ser registrados en las especificaciones.

Atributo al que ayuda a calcular: seguridad

Rango: $0 \leq X \leq 1$, entre más cerca a 1, mas auditable

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 185

Nombre en español: Capacidad de control de acceso – interno

Nombre en inglés: Access controllability

Descripción: ¿Qué tan controlable es el acceso al sistema? Cuenta el número de requerimientos de controlabilidad de acceso implementados correctamente como en las especificaciones y compara con el número de requerimientos de controlabilidad de acceso en las especificaciones

Fórmula:

$$X=A/B,$$

A (Conteo): Número requerimientos de controlabilidad de acceso implementados correctamente como en las especificaciones

B (Conteo): Número requerimientos de controlabilidad de acceso en las especificaciones

Atributo al que ayuda a calcular: Seguridad

Rango: $0 \leq X \leq 1$, entre más cerca a 1, mas controlable

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 186

Nombre en español: Capacidad de auditoría de acceso – interno

Nombre en inglés: Data corruption prevention

Descripción: ¿Qué tan completa es la implementación de prevención de corrupcion de datos? Cuenta el número de instancias implementadas de prevención de corrupcion de datos como especificados y compara con el número de instancias de operaciones /accesos especificados en requerimientos como capaz de corromper/destruir los datos

Fórmula:

$$X=A/B,$$

A (Conteo): Número de instancias implementadas de prevención de corrupcion de datos como especificados confirmados en revisión,

B (Conteo): Número de instancias de operaciones/accesos identificados en requerimientos como capaces de corromper/destruir datos.

Atributo al que ayuda a calcular: Seguridad

Rango: $0 \leq X \leq 1$, entre más cerca a 1, más completo

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 187

Nombre en español: Encriptación de datos

Nombre en inglés: Data encryption

Descripción: ¿Qué tan completa es la implementación de encriptación de datos? Cuenta el número de instancias implementadas de encriptación / des encriptación de datos como especificados y compara con el número de instancias de de datos que requieren facilidad de encriptación / des encriptación de datos como en las especificaciones

Fórmula:

$$X=A/B,$$

A (Conteo): Número de instancias implementadas de encriptación / des encriptación de datos como especificados confirmados en revisión,

B (Conteo): Número de instancias que requieren facilidad de encriptación / des encriptación de datos como en las especificaciones,

Atributo al que ayuda a calcular: Seguridad

Rango: $0 \leq X \leq 1$, entre más cerca a 1, más completo

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 188

Nombre en español: Medida de estructura de diseño

Nombre en inglés: Design structure measure

Descripción: La métrica es la suma de los puntajes de los siguientes elementos aplicables dividido el número de elementos aplicables:

1. Diseño organizado de arriba hacia abajo en la moda: Un cuadro jerárquico de módulos del sistema esta usualmente disponible o fácil de construir desde la documentación de diseño. este debe reflejar el concepto aceptado del diseño de arriba hacia abajo. el sistema está organizado en una estructura de árbol jerárquico, cada nivel del árbol representa bajos niveles de descripción en detalle del procesamiento.
2. Funciones no duplicadas: las descripciones de funciones a realizar por cada modulo en diseño y la actual función realizada por el código del

modulo es evaluada para asegurarse que no duplicara por otros módulos

3. Independencia de modulo: El procesamiento hecho dentro de un modulo no depende de la fuente de entrada o el destino de salida. esta regla puede ser aplicada a la descripción de modulo durante el diseño y código del modulo durante la implementación. la medida para este elemento está basada en el número de módulos que no cumplen con esta regla.
4. El procesamiento de un modulo no depende de un procesamiento anterior: el procesamiento hecho dentro de un modulo no es dependiente de los conocimientos o resultados de un procesamiento anterior, por ejemplo, la primera vez a través del modulo, la enésima vez a través del modulo, etc. esta regla es aplicada según lo indicado en diseño e implementación.
5. Cada descripción de modulo incluye entrada, salida, procesamiento, limitaciones: Documentación que describe la entrada, salida, procesamiento, y limitaciones para cada modulo es desarrollada durante diseño y está disponible durante implementación. la medida de este elemento está basada en el número de módulos que no tienen esta información documentada.
6. Cada modulo tiene un acceso único, existencia única: Determinación del número de módulos que violan esta regla en diseño e implementación pueden ser hechas, y es la base de esta métrica.
7. Dato no global: Esta es una medida binaria que identifica la complejidad adicional de un sistema por el uso de datos globales. si no existen datos globales, esta medida es 1, si existen datos globales, es 0

Fórmula: A nivel de sistema:

$$X=A/B,$$

A= \sum ai, (Para i de 1 a 4) ai: puntajes de los elementos aplicables,
B (conteo): Número de elementos aplicables.

ELEMENTOS:

- (1),(2) y (7): seguir la definición,
- (3),(4),(5) y (6) usar la formula $Y=(1 - C/D)$,
C (Conteo): Número de módulos que violan la regla,
D (Conteo): Número total de módulos

Atributo al que ayuda a calcular: Simplicidad

Granularidad: sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica en disciplinas de diseño, implementación

MÉTRICA 189

Nombre en español: Uso de lenguaje Estructurado o procesador

Nombre en inglés: Use of structured language or procesor

Descripción: La métrica es una medida binaria de existencia (1) o ausencia (0) de lenguaje de código Estructurado. Un lenguaje Estructurado o un preprocesador de lenguaje Estructurado proporcionan construcciones similares a las declaraciones IFTHENELSE, DOWHILE, DOUNTIL, y CASE asociadas con programación estructurada

Fórmula: A nivel de sistema:

$$X=A/B,$$

$$A=\sum ai,$$

ai: puntajes de los módulos,

B(conteo): Número de módulos.

Atributo al que ayuda a calcular: Simplicidad

Granularidad: Sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica en disciplina de, implementación

MÉTRICA 190

Nombre en español: Medida de Complejidad

Nombre en inglés: Complexity measure

Descripción: Esta métrica puede medirse desde la representación de diseño (por ejemplo, diagramas de flujo) y el código automáticamente. La ruta de análisis de flujo y variables de información colocada/usada a lo largo de cada ruta utilizada. Una variable que considera 'viva' en caso de que un nodo puede ser usado de nuevo por ese camino en el programa. La medida de Complejidad esta basada en la suma de las "vidas" de todas las variables a lo largo de todas las rutas en el programa. normalizan dividiéndolas por la máxima Complejidad del programa (todas las variables viven a lo largo de todas las rutas). (Para cada modulo véase métrica 188)

Fórmula:

$$X=A/B,$$

$$A=\sum ai,$$

ai: medida de Complejidad para cada modulo,

B (conteo): Número de módulos.

Atributo al que ayuda a calcular: Simplicidad

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica en disciplinas de diseño, implementación

MÉTRICA 191

Nombre en español: Medida de simplicidad de técnicas de codificación

Nombre en inglés: Measure of coding simplicity

Descripción: La métrica de nivel de sistema es una cantidad promedio de todas las medidas de módulos para el sistema. La medida de módulo es la suma de los puntajes de los siguientes elementos aplicables dividido el número de elementos aplicables:

1. Módulo de flujo de arriba a abajo: esta es una medida binaria del flujo lógico de un módulo. si fluye de arriba abajo, le da un valor de 1, si no es 0
2. Expresiones booleano negativo o booleano compuesto complejo usadas: expresiones compuestas involucran dos o más operadores booleanos y la negación puede ser a menudo evitada. Este tipo de expresiones añaden Complejidad al módulo. La medida esta basada en el número de estas expresiones complicadas por declaraciones ejecutables en el módulo
3. Salto dentro y fuera de los bucles: bucles dentro de un módulo pueden tener una entrada y una salida. esa medida está basada en el número de bucles que cumplen con esta regla dividido por el número total de bucles.
4. Índice de bucle modificado: Modificación del índice de un bucle no sólo complica la lógica de un módulo, sino causa graves problemas mientras la depuración. Esta medida esta basada en el número de bucles que modifican los índices, dividido por el número total de bucles.
5. El Módulo no es auto modificable: Si un módulo tiene la capacidad de modificar su lógica de procesamiento es muy difícil de reconocer en qué estado esta cuando produce un error. Además, el análisis de la lógica es más difícil. Esta medida enfatiza la complejidad añadida de la auto modificación de módulos.
6. Todos los argumentos pasados a un módulo son paramétricos: Esta es una medida binaria, 1 si todos los parámetros son paramétricos, 0 si todos no lo son. Esta medida esta basada en los problemas potenciales que pueden surgir si constantes o datos globales son utilizados como argumentos.
7. Número de etiquetas declaradas: Esta medida esta basada en la premisa de que la declaración que más etiquetas son añadidas a un módulo más complejo a entender.
8. Nombres únicos para variables: Esta es una medida binaria que le asigna un 1 si son utilizados nombres únicos, y un 0 si no lo son.
9. Uso único de variables: Una variable que utiliza para un solo propósito, es decir, de una manera. Esta medida es una medida binaria, 1 si son utilizados las variables de una sola forma y 0 si son utilizados para múltiples propósitos.

10. Expresiones de modo no mixto: Si las expresiones de modo de mezcla son utilizados introduce una mayor Complejidad. Esta medida es un 1 si las expresiones que no están en modo de mezcla son utilizados en un módulo, y un 0 si las expresiones del modo de mezcla son utilizados.
11. Nivel de anidación: Cuanto mayor sea el nivel de anidación de las decisiones o bucles en un módulo, mayor es la Complejidad. La medida es la inversa de el máximo nivel de anidamiento
12. Número de ramificaciones: a más caminos o ramificaciones que encuentra en un módulo, mayor es la Complejidad. Esta medida esta basada en el número de declaraciones de decisión para declaraciones ejecutables.
13. Número de GOTO: Mucho esta escrito en la literatura acerca de las virtudes de evitar el GOTO. Esta medida esta basada en el número de declaraciones GOTO para declaraciones ejecutables.
14. No existe código de extraños: Esta es una medida binaria que es 1 si no existe código de extraños y 0 si existe. codigo de extraños es el código que no es funcional o no puede ser ejecutado.
15. Variable mixta en un modulo: Desde el punto de vista de la simplicidad, las variables locales son mucho mejores que las variables globales. Esta medida es la proporción de variables internas (locales) con el total de variables dentro del modulo (internas (locales) más externa (global)).
16. Densidad de variable: El uso de más variables en un módulo, mayor es la complejidad de dicho módulo. Esta medida esta basada en el número de usos de variable en un módulo divididos por el máximo posible de usos.

Fórmula: A nivel de sistema:

$$X=A/B;$$

A: Puntaje total de elementos aplicables,

B: Número de elementos aplicables;

Fórmula: A nivel de modulo:

$$Y=C/D,$$

$$C=\sum a_i,$$

a_i : medida de simplicidad de codigo para cada modulo,

D: Número de módulos.

ELEMENTOS:

- (1), (5), (6), (8), (9), (10) y (14): Seguir la definición
- (2): $M=(1 - Q/Z)$,
Q: Número de arriba,
Z: Número declaraciones ejecutables,
- (3): $M=Q/Z$,

- Q: bucles de única entrada/ única salida,
Z: Número total de bucles
- (4): $M=Q/Z$,
Q: Índices de bucles modificados,
Z: Número total de bucles.
- (7): $M=(1 - Q/Z)$,
Q: Número de etiquetas,
Z: Número declaraciones ejecutables,
- (11): $M=1/Z$,
Z: máximo nivel de anidación
- (12): $M=(1 - Q/Z)$,
Q: Número de ramificaciones,
Z: Número declaraciones ejecutables
- (13): $M=(1 - Q/Z)$,
Q: Número de declaraciones GOTO,
Z: Número declaraciones ejecutables
- (15): $M= Q/Z$,
Q: Número de variables internas,
Z: Número total de variables
- (17): $M=(1 - Q/Z)$,
Q: Número de variables,
Z: Número declaraciones ejecutables

Atributo al que ayuda a calcular: Simplicidad

Granularidad: Sistema, modulo

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica a disciplina de implementación, aplicado a modulo de primer nivel

MÉTRICA 192

Nombre en español: Referencias cruzadas relacionando módulos a requerimientos

Nombre en inglés: Cross reference relating modules to requirements

Descripción: Durante el diseño, la identificación de cuales requerimientos detallados son satisfechos en el diseño de un modulo son documentados. La matriz de trazabilidad es un ejemplo de cómo puede hacerse esto. Durante la implementación, cuales requerimientos detallados comienzan a satisfacerse por el modulo de implementación identificado. Alguna forma de notación automatizada, comentarios de prologo o insertando comentarios, es usado para proveer referencia cruzada. la métrica es el número de requisitos de trazabilidad detallada dividida por el número total de requerimientos detallados

Fórmula: A nivel de sistema:

$$X=A/B,$$

A (Conteo): Número de requisitos de trazabilidad detallada,

B (Conteo): Número total de requerimientos (El valor es tomado de la cuenta en cada fase)

Atributo al que ayuda a calcular: Trazabilidad

Granularidad: sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica en disciplinas de diseño, implementación

MÉTRICA 193

Nombre en español: Utilización de dispositivos de E/S

Nombre en inglés: I/O Devices utilisation

Descripción: ¿Esta el uso del dispositivo de I/O demasiado alto, causando interferencia? Ejecute concurrentemente un largo número de tareas. Registrar la utilización del dispositivo de I/O, y comparar con los objetivos diseñados

Fórmula:

$$X=A/B$$

A (Tiempo): Tiempo de ocupación de dispositivos de I/O,

B (Tiempo): Tiempo especificado el cual es diseñado para ocupar dispositivos I/O. $0 \leq X \leq 1$, entre más cercano y menor que 1 es mejor

Atributo al que ayuda a calcular: Uso de recursos

Rango: $0 \leq X \leq 1$, entre más cercano y menor que 1 es mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 194

Nombre en español: Límites de carga de E/S

Nombre en inglés: I/O loading limits

Descripción: ¿Cuál es el límite absoluto sobre utilización de I/O en realizar una función? Calibrar la condición de prueba. Simular una condición por la cual el sistema alcance una situación de máxima carga. Ejecutar la aplicación y monitorear los resultados.

Fórmula:

$$X=A_{max}/R_{max},$$

A_{max} (Conteo)=MAX (A_i) (para $i=1$ hasta N)

R_{max} (Conteo): máximo requerido de mensajes I/O.

MAX (A_i): Número máximo de mensajes de I/O desde la 1ª hasta la i -ésima evaluación.

N (Conteo): Número de evaluaciones.

Atributo al que ayuda a calcular: uso de recursos

Rango: $0 \leq X$, entre más corto 1 es mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 195

Nombre en español: Cociente de cumplimiento medio de E/S

Nombre en inglés: Mean I/O fulfillment ratio

Descripción: ¿Cuál es el número promedio de mensajes de errores relacionados a I/O y fallas sobre una duración especificada de tiempo y una utilización especificada? Calibrar la condición de prueba para la cual el sistema alcanza una situación de máxima carga. Ejecutar la aplicación y registrar el número de errores debidos a fallas de I/O y advertencias.

Fórmula:

$$X = A_{\text{mean}} / R_{\text{mean}},$$

A_{mean} (Conteo) = $\sum (A_i) / N$,

R_{mean} (Conteo): Número medio requerido de mensajes I/O,

A_i (Conteo): Número de mensajes de error I/O para la evaluación i -ésima, N (Conteo): Número de evaluaciones

Atributo al que ayuda a calcular: uso de recursos

Rango: $0 \leq X$, entre más corto 1 es mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 196

Nombre en español: Tiempo de espera de usuario para utilización de dispositivos de E/S

Nombre en inglés: User waiting time of I/O devices utilisation

Descripción: ¿Cuál es el impacto del dispositivo de I/O sobre los tiempos de espera del usuario? Ejecute concurrentemente una amplia cantidad de tareas y mide lo tiempo de espera de usuario como un resultado de operación de dispositivo I/O.

Fórmula:

T = Tiempo gastado de espera para operación de dispositivo I/O.

Nota: recomienda que el tiempo máximo y distribuido sea investigado para varios casos de prueba u operación, por que las medidas tienden fluctuantes por condiciones de uso.

Atributo al que ayuda a calcular: uso de recursos

Rango: $0 < T$, Entre más corto es mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 197

Nombre en español: Utilización máxima de memoria

Nombre en inglés: Maximum memory utilisation

Descripción: ¿Cuál es el límite absoluto sobre la memoria requerida en Cumplir una función? Calibrar la condición de prueba. Simular una condición para la cual el sistema alcanza una situación de carga máxima. Ejecutar la aplicación y monitorear los resultados.

Fórmula:

$$X=A_{\max}/R_{\max},$$

A_{\max} (Conteo)= MAX (A_i) (para $i=1$ hasta N),

R_{\max} (Conteo): Relación de memoria máxima requerida.

MAX (A_i): Número máximo de mensajes de error relacionados a memoria desde la evaluación i -ésima.

N (Conteo): Número de evaluaciones.

Atributo al que ayuda a calcular: Uso de recursos

Rango: $0 \leq X$, entre más corto 1 es mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 198

Nombre en español: Ocurrencia media de errores de memoria

Nombre en inglés: Mean occurrence of memory error

Descripción: ¿Cuál es el número promedio de mensajes de error y fallos relacionados a memoria sobre un periodo de tiempo determinado y una determinada carga de sistema? Calibrar la condición de prueba. Simular una condición para la cual el sistema alcanza una situación de máxima carga. Ejecutar la aplicación y registrar el número de errores debidos a fallas de memoria y advertencias.

Fórmula:

$$X=A_{\text{mean}}/R_{\text{mean}},$$

A_{mean} (Conteo)= $\sum (A_i)/N$,

R_{mean} (Conteo): Número medio requerido de mensajes de error relacionados a memoria,

A_i (Conteo): Número de mensajes de error relacionados a memoria para la evaluación i -ésima.

N (Conteo): Número de evaluaciones.

Atributo al que ayuda a calcular: Uso de recursos

Rango: $0 \leq X$, entre más corto es mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 199

Nombre en español: Cociente error/tiempo de memoria

Nombre en inglés: Ratio of memory error/time

Descripción: ¿Cuántos errores de memoria fueron experimentados sobre un periodo de tiempo y especificado la utilización de recursos? Calibrar las condiciones de prueba. Simular una condición para la cual el sistema alcanza una situación de máxima carga. Ejecutar la aplicación y registrar el número de errores debidos a fallas de memoria y advertencia.

Fórmula:

$$X=A/T,$$

A (Conteo): Número de mensajes de advertencia o fallas del sistema,

T (Tiempo): Tiempo de operación de usuario durante la observación del usuario

Atributo al que ayuda a calcular: Uso de recursos

Rango: $0 \leq X$, entre más corto es mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 200

Nombre en español: Utilización máxima de transmisión

Nombre en inglés: Maximum transmission utilisation

Descripción: ¿Cuál es el límite absoluto de transmisiones requeridas para cumplir una función? Evaluar que es requerido por el sistema para alcanzar una situación de máxima carga. Simular esa condición. Ejecutar la aplicación y monitorear resultados

Fórmula:

$$X = A_{\max} / R_{\max},$$

A_{\max} (Conteo) = MAX (A_i) (para $i=1$ hasta N),

R_{\max} (Conteo): número máximo requerido de mensajes de error y fallas relacionados a transmisiones.

MAX (A_i): Número máximo de mensajes de error relacionados a transmisión desde la evaluación 1ª hasta la i -ésima.

N (Conteo): Número de evaluaciones. 0

Atributo al que ayuda a calcular: uso de recursos

Rango: $0 \leq X$, entre más corto es mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 201

Nombre en español: Balance de utilización del dispositivo de comunicación

Nombre en inglés: Media device utilisation balancing

Descripción: ¿Cuál es el grado de sincronización entre diferentes medios sobre un periodo de tiempo? Calibrar las condiciones de prueba. Simular una condición para la cual el sistema alcanza una situación de máxima carga de transmisión. Ejecutar la aplicación y registrar el retraso en el procesamiento de diferentes tipos de medio.

Fórmula:

$$X = \text{SyncTime} / T,$$

SyncTime (Tiempo): Tiempo dedicado a los recursos continuos,

T (tiempo): periodo de tiempo requerido durante el cual diferentes medios de comunicación esperan terminar sus tareas con sincronía.

Atributo al que ayuda a calcular: uso de recursos

Rango: Entre más pequeño es el cociente es mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 202

Nombre en español: Ocurrencia media de error de transmisión

Nombre en inglés: Mean occurrence of transmission error

Descripción: ¿Cuál es el número promedio de mensajes de error y fallos relacionados a transmisión sobre un periodo de tiempo determinado y una determinada utilización? Calibrar la condición de prueba. Simular una condición para la cual el sistema alcanza una situación de máxima carga. Ejecutar la aplicación y registrar el número de errores debidos a fallas en la transmisión y advertencias

Fórmula:

$$X = A_{\text{mean}} / R_{\text{mean}},$$

A_{mean} (Conteo): $\sum(A_i) / N$,

R_{mean} (Conteo): número medio requerido de mensajes de error y fallos relacionados a transmisión,

A_i (Conteo): Número de mensajes de error y fallas relacionados a transmisión para la evaluación i -ésima.

N (Conteo): Número de evaluaciones.

Atributo al que ayuda a calcular: Uso de recursos

Rango: $0 \leq X$, entre más corto es mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 203

Nombre en español: Media de error de transmisión por vez

Nombre en inglés: (c) Mean occurrence of transmission error

Descripción: ¿Cuál es el número promedio de mensajes de error y fallos relacionados a transmisión sobre un periodo de tiempo determinado y una determinada utilización? Calibrar la condición de prueba. Simular una condición para la cual el sistema alcanza una situación de máxima carga. Ejecutar la aplicación y registrar el número de errores debidos a fallas en la transmisión y advertencias

Fórmula:

$$X = A_{\text{mean}} / R_{\text{mean}},$$

A_{mean} (Conteo): $\sum(A_i) / N$,

R_{mean} (Conteo): número medio requerido de mensajes de error y fallos relacionados a transmisión,

A_i (Conteo): Número de mensajes de error y fallas relacionados a transmisión para la evaluación i -ésima.

N (Conteo): Número de evaluaciones. $0 \leq X$, entre más corto es mejor.

Atributo al que ayuda a calcular: Uso de recursos

Rango: $0 \leq X$, entre más corto es mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 204

Nombre en español: Utilización de capacidad de transmisión

Nombre en inglés: Transmission capacity utilisation

Descripción: ¿Es el sistema software capaz de realizar tareas dentro de una capacidad de transmisión esperada? Ejecute concurrentemente tareas específicas con múltiples usuarios, observe la capacidad de transmisión y compare uno especificado.

Fórmula:

$$X=A/B,$$

A (Tamaño): Capacidad de transmisión

B (Tamaño): Capacidad de transmisión especificada la cual es designada para ser usada por el software durante la ejecución

Atributo al que ayuda a calcular: Uso de recursos

Rango: entre más pequeño y cerca a 1 es mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 205

Nombre en español: Utilización de E/S

Nombre en inglés: I/O utilization

Descripción: ¿Cuál es la utilización estimada de I/O para completar una tarea específica? Estimar los requerimientos de utilización de I/O para la aplicación

Fórmula:

X (Tamaño)=Número de buffers (Calculados o simulados).

Atributo al que ayuda a calcular: uso de recursos

Rango: Entre mas corto es el cociente mejor

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 206

Nombre en español: Densidad de utilización de mensajes E/S

Nombre en inglés: I/O utilization Menssage density

Descripción: ¿Cuál es la densidad de mensajes relacionados a utilización de I/O en líneas de código responsable en hacer llamadas a sistema?. Cuenta el número de errores relativos a fallas de I/O y fallas y lo compara al número estimado de líneas de código responsables en llamadas a sistema

Fórmula:

$$X=A/B,$$

A (Conteo): Número de mensajes de error relacionados a I/O.

B (Conteo): Número de líneas de código directamente relacionadas a llamadas a sistema.

Atributo al que ayuda a calcular: uso de recursos

Rango: el más grande de mejor

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 207

Nombre en español: Utilización de memoria

Nombre en inglés: Memory utilization

Descripción: ¿Cuál es el tamaño de memoria estimado que el producto ocupara para completar una tarea específica? Estimar el requerimiento de memoria

Fórmula:

X (Tamaño)=tamaño en bytes (Calculados o simulados).

Rango: Entre mas corto es el cociente mejor

Granularidad:

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 208

Nombre en español: Densidad de utilización de mensajes de memoria

Nombre en inglés: Memory utilization Menssage density

Descripción: ¿Cuál es la densidad de memoria relacionada a utilización de memoria en líneas de código responsable en hacer llamadas a sistema? Cuenta el número de mensajes de error relativos a fallas y advertencias de memoria y lo compara al número estimado de líneas de código responsables en llamadas a sistema

Fórmula:

$$X=A/B,$$

A (Conteo): Número de mensajes de error relacionados a memoria.

B (Conteo): Número de líneas de código directamente relacionadas a llamadas a sistema.

Atributo al que ayuda a calcular: uso de recursos

Rango: Entre más grande mejor

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 209

Nombre en español: Utilización de transmisión

Nombre en inglés: Transmission Utilization

Descripción: ¿Cuál es la cantidad estimada de utilización de recursos de transmisión? Estimar los requerimientos de utilización de recursos de transmisión por el estimado de volúmenes de transmisión

Fórmula:

X(Tiempo)=bites/time (Calculados o simulados).

Atributo al que ayuda a calcular: Uso de recursos

Rango: Entre más pequeño es mejor

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 210

Nombre en español: Comunicatividad 1

Nombre en inglés: Comunicativeness 1

Descripción: ¿El programa suministra etiquetas para todas las entradas solicitadas?

Atributo al que ayuda a calcular: Comunicatividad

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 211

Nombre en español: Comunicatividad 2

Nombre en inglés: Comunicativeness 2

Descripción: ¿El programa reconoce el final de entradas sin que el usuario sea solicitado para contar y especifica el número de entradas (e.g., reconocimiento de un campo de entrada con formato especial, carácter, valor, fin de archivo, etc.)?

Atributo al que ayuda a calcular: Comunicatividad

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 212

Nombre en español: Comunicatividad 3

Nombre en inglés: Comunicativeness 3

Descripción: ¿El programa contiene provisiones para desplegar identificación o descripción de casos de prueba sobre todos los resultados de pruebas?

Atributo al que ayuda a calcular: Comunicatividad

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 213

Nombre en español: Comunicatividad 4

Nombre en inglés: Comunicativeness 4

Descripción: ¿El programa tiene la capacidad para desplegar mensajes de error claros y útiles si éste ocurre?

Atributo al que ayuda a calcular: Comunicatividad

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 214

Nombre en español: Comunicatividad 5

Nombre en inglés: Comunicativeness 5

Descripción: ¿El programa tiene Facilidades para ejecutar múltiples casos de prueba sin especificación redundante de valores de entrada sin cambiar?

Atributo al que ayuda a calcular: Comunicatividad

Referencias: Boehm [17]
Espectro: Oscuro

MÉTRICA 215

Nombre en español: Comunicatividad 6
Nombre en inglés: Communicativeness 6
Descripción: ¿El programa contiene provisiones para desplegar resultados intermedios a peticiones?
Atributo al que ayuda a calcular: Comunicatividad
Referencias: Boehm [17]
Espectro: Oscuro

MÉTRICA 216

Nombre en español: Comunicatividad 7
Nombre en inglés: Communicativeness 7
Descripción: ¿Son identificadas todas las salidas del programa con etiquetas descriptivas?
Atributo al que ayuda a calcular: Comunicatividad
Referencias: Boehm [17]
Espectro: Oscuro

MÉTRICA 217

Nombre en español: Comunicatividad 8
Nombre en inglés: Communicativeness 8
Descripción: ¿El programa provee, para aceptación, entradas en formularios con campos libres (e.g., no requiere identificación conteo de columnas)?
Atributo al que ayuda a calcular: Comunicatividad
Referencias: Boehm [17]
Espectro: Oscuro

MÉTRICA 218

Nombre en español: Comunicatividad 9
Nombre en inglés: Communicativeness 9
Descripción: ¿El programa contiene la capacidad para seguir y desplegar flujos de control lógicos?
Atributo al que ayuda a calcular: Comunicatividad
Referencias: Boehm [17]
Espectro: Oscuro

MÉTRICA 219

Nombre en español: Comunicatividad 10
Nombre en inglés: Communicativeness 10

Descripción: ¿Están identificadas las salidas principales del programa y desplegadas en forma que permitan resumir la evaluación de resultados de prueba?

Atributo al que ayuda a calcular: Comunicatividad

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 220

Nombre en español: Comunicatividad 11

Nombre en inglés: Communicativeness 11

Descripción: ¿El programa provee paginación lógica de resultados impresos para soportar examen visual?

Atributo al que ayuda a calcular: Comunicatividad

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 221

Nombre en español: Medida de interface de entrada de usuario

Nombre en inglés: User input interface measure

Descripción: La métrica es la suma de los puntajes dados por los siguientes elementos aplicables divididos por el número de elementos aplicables:

1. (1)Valores definidos por defecto: (diseño e implementación) Un método de minimizar la cantidad requerida es proporcionar valores predeterminados. Esta medida, aplicada durante el diseño e implementación, esta basada en el número de predeterminados permitido dividido por el número total de parámetros de entrada
2. Formatos de entrada uniformes: a mayor número de formatos de entrada es la más difícil usar el sistema. Esta medida esta basada en el número total de formatos de entrada
3. Cada registro de entrada auto identificable: Registros de entrada que tienen códigos de auto identificación de realzan la exactitud de las entradas de usuario. Esta medida esta basada en el número de registros de entrada que no son de auto identificación, dividido por el número total de registros de entrada. esto Aplica en el diseño y aplicación
4. La entrada puede verificarse por el usuario antes de la ejecución: (diseño e implementación) La capacidad, viendo la entrada a petición o haciendo eco de las entradas automáticamente, permite al usuario para verificar sus entradas antes del procesamiento. Esta es una medida de la existencia en diseño e implementación de esta capacidad.
5. Entrada terminada por definirse explícitamente el fin lógico de la entrada:(diseño e implementación) El usuario no debería tener que

proporcionar una cuenta de tarjetas de entrada. Esta es una medida binaria de la elaboración e implementación de esta capacidad.

6. Provisión para especificar entradas desde diferentes medios: La flexibilidad de entrada debe ser decidida durante la fase de análisis de requerimientos y en seguida a través de diseño e implementación. Esta es una medida binaria de la existencia de la consideración de esta capacidad durante las tres fases.

Fórmula: A nivel de sistema:

$$X=A/B,$$

A: Puntaje total de los elementos aplicables,

B: Número de elementos aplicables.

ELEMENTOS:

- (1): $Y=C/D$,
C: Número de predeterminados,
D: Número total de parámetros; (
- 2): $Y=1/D$,
D: Número de entradas de diferentes formatos de registro;
- (3): $Y=1-(C/D)$,
C: Número que no son auto identificables,
D: Número total de registros de entradas;
- (4),(5) y(6)Seguir la definición

Atributo al que ayuda a calcular: Comunicatividad

Granularidad: Sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica a disciplinas de análisis, diseño, Implementación en el nivel del sistema

MÉTRICA 222

Nombre en español: Medida de interface de salida de usuario

Nombre en inglés: User output interface measure

Descripción: La métrica es la suma de los puntajes dados por los siguientes elementos aplicables divididos por el número de elementos aplicables:

1. (1)Controles de salida selectivos: La existencia de un requisito de diseño, e implementación de controles de salida selectiva indica en la presente medida binaria. los controles selectivos incluye la elección de de salidas específicas, formatos de salida, cantidad de la producción, etc.
2. (2)Salidas tienen etiquetas orientadas a usuario descriptivas únicas: (diseño e implementación) Esta es una medida binaria de la diseño e implementación de etiquetas de salida única. Además entre otras cosas, las etiquetas serán descriptivas para el usuario. Esto incluye no

sólo las etiquetas que son utilizados para hacer referencia a un informe de salida, sino también el título, los encabezamientos de las columnas, etc., dentro de ese informe.

3. (3)Salidas tienen unidades orientadas al usuario: (diseño e implementación) Esta es una medida binaria que extiende de (2) en la salida individual.
4. (4)Etiquetas de salida uniforme: (diseño e implementación) Esta medida corresponde a métrica 221 (2) anterior y es la inversa del número de diferentes formatos de salida.
5. (5)Grupos lógicos de salidas separadas por exanimación del usuario: (diseño e implementación) Utilización de parte superior de la página, líneas en blanco, las líneas de asteriscos, etc., provistas para una fácil identificación de agrupar lógicamente salidas. Esta medida binaria identifica si estas técnicas son utilizados durante el diseño e implementación.
6. (6)Relación entre mensajes de error y salidas es ambigua: (diseño e implementación) esta es una medida binaria aplicada durante el diseño y la implementación que identifica si los mensajes de error son relacionados directamente con la salida
7. (7)Provisión para la reorientación de salidas a los diferentes medios: Esta es una métrica binaria que identifica si da consideración a la capacidad de redirigir la salida a los diferentes medios durante el análisis de requisitos, diseño e implementación.

Fórmula: A nivel de sistema:

$$X=A/B,$$

A: Puntaje total de los elementos aplicables,

B: Número de elementos aplicables.

ELEMENTOS:

- (1),(2),(3),(5),(6) y (7):Seguir la definición;
- (4): $Y=1/D,$

D: Número de de diferentes formatos de salida

Atributo al que ayuda a calcular: Comunicatividad

Granularidad: sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica a disciplinas de diseño, Implementación en el nivel del sistema

MÉTRICA 223

Nombre en español: Interacción atractiva – externo

Nombre en inglés: Attractive interaction

Descripción: ¿Qué tan atractiva es la interfaz para el usuario? Cuestionarios para usuarios

Fórmula:

Cuestionario para asegurar el atractivo de la interfaz para usuarios, después de la experiencia de uso. (Conteo)

Atributo al que ayuda a calcular: Atractivo

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 224

Nombre en español: Personalización de apariencia de interfaz- externo

Nombre en inglés: Interface appearance customisability

Descripción: ¿Cuál es la proporción de elementos de interfaz que pueden ser personalizados en apariencia para la satisfacción del usuario? Test de conducta de usuario y observación de conducta de usuario

Fórmula:

$$X=A/B,$$

A (conteo): Número de elementos de interfaz personalizados en apariencia para satisfacción del usuario.

B (conteo): número de elementos de interfaz que el usuario desea personalizar.

Atributo al que ayuda a calcular: Atractivo

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 225

Nombre en español: Interacción atractiva – interno

Nombre en inglés: Attractive interaction

Descripción: ¿Cuan atractiva es la interface para el usuario? Cuestionario a usuarios

Fórmula:

Cuestionario para asegurar la atraktividad de la interfaz para usuarios, toma cuenta de atributos como color y diseño grafico. NOTA: temas que pueden contribuir al atractivo incluyen: alineación de ítems (Vertical y horizontal), agrupamiento, uso de colores, tamaño de graficas apropiado y razonable, uso de espacios en blanco/separadores/bordes, animaciones, tipografía, e interfaces 3D

Atributo al que ayuda a calcular: atractivo

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 226

Nombre en español: Personalización de apariencia de interfaz –interno

Nombre en inglés: Interface appearance customisability

Descripción: ¿Qué proporción de elementos de interfaz de usuario pueden ser personalizados en apariencia? Inspección (Por experto)

Fórmula:

$$X=A/B,$$

A (conteo): Número de tipos de elementos de interfaz que pueden ser personalizados.

B (conteo): número de tipos de elementos de interfaz.

Atributo al que ayuda a calcular: Atractivo

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 mejor

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 227

Nombre en español: Lista de chequeo de entrenamiento

Nombre en inglés: Training checklist

Descripción: La métrica es la suma de los puntajes dados por los siguientes elementos aplicables, dividido por el número de elementos aplicables:

1. Materiales de Planeación/entrenamiento de enseñanza desarrollados para operadores, usuarios finales, mantenedores: (implementación) esta es una medida binaria de si esta tipo de documentación es dada durante la fase de implementación
2. Ejercicios simulados realistas suministrados: esta es una medida binaria de si los ejercicios que representan el ambiente operacional, son desarrollados durante la fase de implementación para uso en entrenamiento
3. Suficiencia de 'ayuda' y diagnostico de información disponible en línea: esta es una medida binaria de si la capacidad de ayudar al operador a la familiarización con el sistema ha sido diseñada y construida en el sistema. provisión de una lista de comandos legales o una lista de pasos secuenciales incluidos en el proceso son ejemplos.

Fórmula: A nivel de sistema:

$$X=A/B,$$

A: Puntaje total de los elementos aplicables,

B: Número de elementos aplicables.

Atributo al que ayuda a calcular: Formación

Granularidad: Sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica a disciplinas de diseño, Implementación en el nivel del sistema

MÉTRICA 228

Nombre en español: Fácil aprendizaje de función

Nombre en inglés: Ease of function learning

Descripción: ¿Cuánto le toma al usuario aprender a utilizar una función? Test de conducta de usuario y observación de comportamiento de usuario

Fórmula:

T= Tiempo medio que le toma aprender a usar una función correctamente.

Atributo al que ayuda a calcular: Facilidad de aprendizaje

Rango: 0<T, entre más pequeño mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 229

Nombre en español: Fácil aprendizaje de realizar una tarea en uso

Nombre en inglés: Ease of learning to perform a task in use

Descripción: ¿Cuánto le toma al usuario aprender como desempeñar una tarea específica eficientemente? Observación de comportamiento de usuario desde cuando empieza a aprender hasta que comience a operar eficientemente

Fórmula:

T= \sum de tiempo de operación de usuario hasta que logre desempeñar la tarea especificada dentro de un corto periodo de tiempo.

Atributo al que ayuda a calcular: Facilidad de aprendizaje

Rango: 0<T, entre más pequeño mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 230

Nombre en español: Efectividad de la documentación de usuario y sistema de ayuda

Nombre en inglés: Effectiveness of the user documentation and/or help system

Descripción: ¿Cuál es la proporción de tareas que pueden ser completadas correctamente después de usar documentación y/o ayuda del sistema? Test de conducta de usuario y observación de conducta de usuario. Cuenta el número de tareas completadas satisfactoriamente después de documentación y/o ayuda de acceso en línea y compara con el número total de tareas probadas

Fórmula:

$$X=A/B,$$

A (conteo): Número de tareas completadas satisfactoriamente después de documentación y/o ayuda de acceso en línea.

B (conteo): número total de tareas probadas.

Atributo al que ayuda a calcular: Facilidad de aprendizaje

Rango: X (0<=X<=1): entre más cerca a 1 mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 231

Nombre en español: Efectividad de la documentación de usuario y/o sistema de ayuda en uso

Nombre en inglés: Effectiveness of the user documentation and/or help system in use

Descripción: ¿Cuál es la proporción de funciones que pueden ser usadas correctamente después de leer la documentación usar sistemas de ayuda? Observación de conducta de usuario. Cuenta el número de funciones usadas correctamente después de leer documentación o usar sistemas de ayuda y compara con el número total de funciones

Fórmula:

$$X=A/B,$$

A (conteo): Número de funciones que pueden ser usadas.

B (conteo): número total de funciones provistas.

Atributo al que ayuda a calcular: Facilidad de aprendizaje

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 232

Nombre en español: Accesibilidad a ayuda

Nombre en inglés: Help accessibility

Descripción: ¿Cuál es la proporción de temas de ayuda que pueden ser usados localmente? Test de conducta de usuario y observación de conducta de usuario. Cuenta el número de tareas para las cuales la ubicación de la ayuda en línea es correcta y compara con el número total de tareas probadas

Fórmula:

$$X=A/B,$$

A (conteo): Número de tareas para las cuales la ubicación de la ayuda en línea es correcta.

B (conteo): número total de tareas probadas.

Atributo al que ayuda a calcular: Facilidad de aprendizaje

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 233

Nombre en español: Frecuencia a ayuda

Nombre en inglés: Help frequency

Descripción: ¿Cuán frecuente un usuario tiene acceso a ayuda para aprender a completar su trabajo? Test de conducta de usuario y observación de conducta de usuario. Cuenta el número de casos que un usuario acede a ayuda para completar su trabajo

Fórmula:

$$X=A/B,$$

A (conteo): Número de accesos a ayuda hasta que un usuario complete su tarea.

B (conteo): número total de tareas probadas

Atributo al que ayuda a calcular: Facilidad de aprendizaje

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 234

Nombre en español: Completitud de documentación de usuario y/o Facilidad de ayuda

Nombre en inglés: Completeness of user documentation and/or help facility

Descripción: ¿Cuál es la proporción de funciones que son descritas en la documentación de usuario y/o Facilidad de ayuda? Test de conducta de usuario y observación de conducta de usuario. Cuenta el número de tareas para las cuales la ubicación de la ayuda en línea es correcta y compara con el número total de tareas probadas

Fórmula:

$$X=A/B,$$

A (conteo): Número de funciones descritas.

B (conteo): número total funciones suministradas.

Atributo al que ayuda a calcular: Facilidad de aprendizaje

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 más completo

Referencias: ISO 9126 [15]:

Espectro: Gris

MÉTRICA 235

Nombre en español: Adecuación funcional - externo

Nombre en inglés: Functional Adequacy

Descripción: ¿Cuan adecuadas son las funciones de evaluación? Número de funciones adecuadas para realizar las tareas específicas comparando el número de funciones evaluadas

Fórmula:

$$X=1-A/B,$$

A (conteo): Número de funciones en donde detectan problemas en la evaluación.

B (conteo): número funciones evaluadas.

Atributo al que ayuda a calcular: Idoneidad

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 236

Nombre en español: Completitud de la implementación funcional – externo

Nombre en inglés: Functional implementación completeness

Descripción: ¿Qué tan completa esta la implementación de acuerdo a las especificaciones de requerimientos? Hacer pruebas funcionales (pruebas de caja negra) del sistema de acuerdo a las especificaciones de requisitos. Cuenta el número de funciones faltantes detectadas en la evaluación y compara con el número de funciones descritas en las especificaciones de requerimientos.

Fórmula:

$$X=1-A/B,$$

A (conteo): Número de funciones faltantes detectadas en la evaluación.

B (conteo): número funciones descritas en la especificación de requerimientos.

Atributo al que ayuda a calcular: Idoneidad

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 237

Nombre en español: Cobertura de la implementación funcional – externo

Nombre en inglés: Functional implementation Coverage

Descripción: ¿Cuan correcta es la implementación funcional? Hacer pruebas funcionales (pruebas de caja negra) del sistema de acuerdo a las especificaciones de requisitos. Cuenta el número de funciones implementadas incorrectamente o faltantes detectadas en la evaluación y compara con el número de funciones descritas en las especificaciones de requerimientos. Cuenta el número de funciones que están completas frente a las que no lo están

Fórmula:

$$X=1-A/B,$$

A (conteo): Número de funciones implementadas incorrectamente o faltantes detectadas en la evaluación.

B (conteo): número funciones descritas en la especificación de requerimientos.

Atributo al que ayuda a calcular: Idoneidad

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 238

Nombre en español: Estabilidad (volatilidad) de la especificación funcional – externo

Nombre en inglés: Functional specification stability (volatility)

Descripción: ¿Cuan estable es la especificación funcional después de entrada en operación? Cuenta el número de funciones descritas en las especificaciones funcionales que han tenido que ser cambiadas después de

que el sistema es puesto en operación y compara con el número total de funciones descritas en la especificación de requisitos

Fórmula:

$$X=1-A/B,$$

A (conteo): Número de funciones cambiadas después de entrar en operación a partir de entrar en operación.

B (conteo): número funciones descritas en la especificación de requerimientos.

Atributo al que ayuda a calcular: Idoneidad

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 239

Nombre en español: Adecuación funcional – interno

Nombre en inglés: Functional Adequacy

Descripción: ¿Cuan adecuadas son las funciones de chequeo? Cuenta el número de funciones implementadas que son adecuadas para realizar las tareas específicas, entonces mide el coeficiente de las funciones implementadas. Lo siguiente puede ser medido:- Todas la partes de especificaciones de diseño, - Módulos/partes completadas de productos software

Fórmula:

$$X=1-A/B,$$

A (conteo): Número de funciones en las cuales los problemas son detectados en la evaluación.

B (conteo): número funciones chequeadas.

Atributo al que ayuda a calcular: Idoneidad

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 más adecuado

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 240

Nombre en español: Completitud de la implementación funcional – interno

Nombre en inglés: Functional implementación completeness

Descripción: ¿Cuan completa es la implementación funcional? Cuenta el número de funciones faltantes detectadas en evaluación y compara con el número de funciones descritas en la especificación de requerimientos

Fórmula:

$$X=1-A/B,$$

A (conteo): Número de funciones faltantes detectadas en la evaluación.

B (conteo): número funciones descritas en la especificación de requerimientos. NOTA: entrada para el proceso de medición son las especificaciones de requerimientos actualizadas. Cualquier cambio

identificado durante el ciclo de vida debe ser aplicado a las especificaciones de requerimientos antes de usarlos en el proceso de medición

Atributo al que ayuda a calcular: Idoneidad

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 más completo

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 241

Nombre en español: Cobertura de la implementación funcional – interno

Nombre en inglés: Functional implementation Coverage

Descripción: ¿Cuan correcta es la implementación funcional? Cuenta el número de implementaciones incorrectas o funciones faltantes y compara con el número de funciones descritas en la especificación de requerimientos. NOTA: Revisión funcional por ítem

Fórmula:

$$X=1-A/B,$$

A (conteo): Número de funciones faltantes detectadas en la evaluación.

B (conteo): número funciones descritas en la especificación de requerimientos. NOTA: entrada para el proceso de medición son las especificaciones de requerimientos actualizadas. Cualquier cambio identificado durante el ciclo de vida debe ser aplicado a las especificaciones de requerimientos antes de usarlos en el proceso de medición

Atributo al que ayuda a calcular: Idoneidad

Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 más completo

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 242

Nombre en español: Estabilidad (volatilidad) de la especificación funcional – interno

Nombre en inglés: Functional specification stability (volatility)

Descripción: ¿Cuan estable es la especificación funcional durante el desarrollo del ciclo de vida? Cuenta el número de funciones cambiadas (adicionadas, modificadas o borradas) durante la fase de desarrollo del ciclo de vida, entonces compara con el número de funciones descritas en la especificación de requerimientos

Fórmula:

$$X=1-A/B,$$

A (conteo): Número de funciones cambiadas durante las fases de desarrollo del ciclo de vida.

B (conteo): número funciones descritas en la especificación de requerimientos.

Atributo al que ayuda a calcular: Idoneidad
Rango: X ($0 \leq X \leq 1$): entre más cerca a 1 más estable
Referencias: ISO 9126 [15]:3
Espectro: Gris

MÉTRICA 243

Nombre en español: Legibilidad 1
Nombre en inglés: Legiility 1
Descripción: ¿Son usadas identificaciones, espacios en blanco, filas y cajas de x ó *, para separar piezas de código?
Atributo al que ayuda a calcular: Legibilidad
Referencias: Boehm [17]
Espectro: Oscura

MÉTRICA 244

Nombre en español: Legibilidad 2
Nombre en inglés: Legiility 2
Descripción: ¿Han sido usados paréntesis o alguna otra técnica para eliminar ambigüedad en el orden o modo de evaluación (real, entero, precisión simple, precisión doble) de expresiones aritméticas?
Atributo al que ayuda a calcular: Legibilidad
Referencias: Boehm [17]
Espectro: Oscura

MÉTRICA 245

Nombre en español: Legibilidad 3
Nombre en inglés: Legiility 3
Descripción: ¿Están en orden ascendente alfanumérico las etiquetas de sentencias?
Atributo al que ayuda a calcular: Legibilidad
Referencias: Boehm [17]
Espectro: Oscura

MÉTRICA 246

Nombre en español: Legibilidad 4
Nombre en inglés: Legiility 4
Descripción: ¿Es pequeña o no la búsqueda a lo largo de las listas requeridas para encontrar todas las sentencias en que ramifica las cuales pueden transferir el control a una etiqueta de sentencia dada?
Atributo al que ayuda a calcular: Legibilidad
Referencias: Boehm [17]
Espectro: Oscura

MÉTRICA 247

Nombre en español: Legibilidad 5

Nombre en inglés: Legiility 5

Descripción: ¿Es consistente el formato y el orden de especificación de sentencias a lo largo de todos los módulos del programa?

Atributo al que ayuda a calcular: Legibilidad

Referencias: Boehm [17]

Espectro: Oscura

MÉTRICA 248

Nombre en español: Legibilidad 6

Nombre en inglés: Legiility 6

Descripción: ¿Es consistente el estilo y formato para los comentarios del programa en todos los módulos del programa?

Atributo al que ayuda a calcular: Legibilidad

Referencias: Boehm [17]

Espectro: Oscura

MÉTRICA 249

Nombre en español: Legibilidad 7

Nombre en inglés: Legiility 7

Descripción: ¿Existe a lo más un parámetro de asignación de valor por línea de código?

Atributo al que ayuda a calcular: Legibilidad

Referencias: Boehm [17]

Espectro: Oscura

MÉTRICA 250

Nombre en español: Legibilidad 8

Nombre en inglés: Legiility 8

Descripción: ¿Existe a lo más una sentencia ejecutable por línea de código?

Atributo al que ayuda a calcular: Legibilidad

Referencias: Boehm [17]

Espectro: Oscura

MÉTRICA 251

Nombre en español: Legibilidad 9

Nombre en inglés: Legiility 9

Descripción: ¿Ha sido empleada alguna regla en el etiquetado de bloques de código funcionalmente distintos?

Atributo al que ayuda a calcular: Legibilidad

Referencias: Boehm [17]

Otros:

MÉTRICA 252

Nombre en español: Evasión de fracaso

Nombre en inglés: Breakdown avoidance, Breakdown ratio

Descripción: ¿Cuan a menudo el producto software causa la descompostura de la producción total del entorno? Cuenta el número de ocurrencia de descomposturas con respecto al número de fallas. Si es bajo la operación, analizar log de la historia de operación de usuario.

Fórmula:

$$X=1- A/B,$$

A (Conteo): Número de descomposturas

B (Conteo): Número de fallas.

Atributo al que ayuda a calcular: Tolerancia al error

Rango: $0 \leq X \leq 1$, entre más cerca a 1 mejor

Referencias: ISO 9126 [15]:2, QUINT2 [58] [59]

Espectro: Gris

MÉTRICA 253

Nombre en español: Evasión de falla -externo

Nombre en inglés: Failure avoidance

Descripción: ¿Cuántos patrones de fallas quedaron bajo control para evitar fallas críticas y graves? Cuenta el número de patrones de fallas evitados y lo compara al número de patrones de fallo a ser considerados

Fórmula:

$$X=A/B,$$

A (Conteo): Número de ocurrencia de fallas serias y críticas evitadas contra casos de prueba de patrones de falla,

B (Conteo): Número de casos de prueba ejecutados de patrones de falla (al menos causando falla) durante pruebas

Atributo al que ayuda a calcular: Tolerancia al error

Rango: $0 \leq X \leq 1$, entre más cerca a 1 mejor, como el usuario puede evitar mas a menudo fallas graves o criticas

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 254

Nombre en español: Evasión de operación incorrecta –externo

Nombre en inglés: Incorrect operation avoidance

Descripción: ¿Cuántas funciones son implementadas con capacidad de evitar operaciones incorrectas? Cuenta el número de casos de prueba de operaciones incorrectas los cuales evitaron causar fallas críticas y graves y lo compara al número de casos de prueba ejecutados en patrones de operación incorrecta a ser conserdados

Fórmula:

$$X=A/B,$$

A (Conteo): Número de ocurrencia de fallas serias y críticas evitadas,

B (Conteo): Número de casos de prueba ejecutados de patrones de falla incorrectos (al menos causando falla) durante pruebas.

Atributo al que ayuda a calcular: Tolerancia al error

Rango: $0 \leq X \leq 1$, entre más cerca a 1 mejor, como la más incorrecta operación del usuario es evitable

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 255

Nombre en español: Evasión de falla - interno

Nombre en inglés: Failure avoidance

Descripción: ¿Cuántos patrones de fallas quedaron bajo control para evitar fallas críticas y graves? Cuenta el número de patrones de fallas evitados y lo compara al número de patrones de fallo a ser considerados

Fórmula:

$$X=A/B,$$

A (Conteo): Número de patrones de falla que tenga evasión en diseño/código,

B (Conteo): Número de patrones de falla a ser considerados.

Atributo al que ayuda a calcular: Tolerancia al error

Rango: $0 \leq X$, donde X es grande la mejor evita fallas

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 256

Nombre en español: Evasión de operación incorrecta – interno

Nombre en inglés: Incorrect operation avoidance

Descripción: ¿Cuántas funciones son implementadas con capacidad de evitar operaciones incorrectas? Cuenta el número de funciones implementadas para evitar causar fallas críticas y graves por operaciones incorrectas y lo compara al número de patrones de operación incorrecta a ser considerados

Fórmula:

$$X=A/B,$$

A (Conteo): Número de funciones implementadas para evitar patrones de operación incorrectos,

B (Conteo): Número de patrones de operación incorrectos a ser considerados.

Atributo al que ayuda a calcular: Tolerancia al error

Rango: $0 \leq X$, donde X es grande evita mejor las operaciones incorrectas.

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 257

Nombre en español: Lista de chequeo de control de tolerancia a error

Nombre en inglés: Error tolerance control check list (Control)

Descripción: La métrica es la suma de los puntajes dados por los siguientes elementos dividido por el número de elementos aplicables:

1. (1)Procesamiento concurrente controlado centralmente: Las funciones que pueden ser usadas concurrentemente son controladas centralmente para proveer comprobación concurrente, bloqueo de escritura/escritura, etc. ejemplos son un manejador de base de datos, manejo de I/O, manejo de error, etc. el control central debe ser considerado en el diseño y después en implementación.
2. (2)Errores reparables y procesamiento continuo: Cuando un error es detectado. la capacidad para corregirlo en línea y continuar procesando, debe estar disponible. un ejemplo es un operador de mensaje que monta la operación incorrecta y continua el procesamiento cuando la grabación correcta es montada. esto puede ser medido en diseño e implementación
3. (3)Cuando una condición de error es detectada, la condición es pasada a la rutina de invocación: la decisión de que hacer sobre un error es hecha a un nivel donde un modulo afectado es controlado. este concepto es construido en el diseño y después en implementación.

Fórmula: A nivel de sistema:

$$X=A/B,$$

A: Puntaje total de los elementos aplicables,

B: Número de elementos aplicables

Atributo al que ayuda a calcular: Tolerancia al error

Granularidad: sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica en disciplinas de diseño, Implementación en el nivel del sistema

MÉTRICA 258

Nombre en español: Recuperación desde la lista de chequeo de datos de entrada impropios

Nombre en inglés: Recovery from improper input data checklist (Input data)

Descripción: La métrica es la suma de los puntajes dados por los siguientes elementos dividido por el número de elementos aplicables:

1. Una declaración definitiva de requerimientos para la tolerancia al error de dato de entrada: la especificación de requerimientos debe identificar las capacidades de tolerancia al error deseadas.
2. Rango de valores (sensatos) para artículos especificados y comprobados: (fases de diseño e implementación solamente) los atributos para cada entrada son comprobados por sensatez. Ejemplos son artículos de comprobación si ellos deben ser numéricos,

alfabéticos, positivos o negativos, de un cierto largo, diferente de cero, etc. estas comprobaciones son especificadas en diseño y existen en código en implementación.

3. Peticiones conflictivas y combinaciones ilegales identificadas y comprobadas: (solo en fases de diseño e implementación) comprobación para ver si los datos de entrada redundantes concuerdan, si las combinaciones de parámetros son razonables, y si las peticiones que son conflictivas pueden ser documentadas en el diseño y existir en el código en implementación.
4. Todas las entradas son comprobadas antes de que el procesamiento comience: (solo en fases de diseño e implementación) la comprobación de entrada no es para detener el primer error encontrado pero es para continuar a través de todas las entradas y reportar los errores. el procesamiento no es para comenzar hasta que los errores son reportados y tampoco las correcciones son hechas o un comando de procesamiento continuo es dado.
5. Determinación que todos los datos están previamente disponibles para procesamiento: para evitar ir a través de varios pasos de procesamiento antes de que los datos de entradas incompletos son encontrados, comprueba por suficiencia de datos de entrada es previo hecha para el comienzo de procesamiento

Fórmula: A nivel de sistema:

$$X=A/B,$$

A: Puntaje total de los elementos aplicables,

B: Número de elementos aplicables

Atributo al que ayuda a calcular: Tolerancia al error

Granularidad: sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica en disciplinas de análisis, diseño, Implementación en el nivel del sistema

MÉTRICA 259

Nombre en español: Recuperación desde la lista de chequeo de fallas computacionales

Nombre en inglés: Recovery from computational failures checklist

Descripción: La métrica es la suma de los puntajes dados por los siguientes elementos dividido por el número de elementos aplicables:

1. Una declaración definitiva de requerimientos para recuperación de posibles fallas computacionales: (solo fase de requerimientos) los

requerimientos para este tipo de capacidad de tolerancia al error son indicados durante la fase de requerimientos

2. Índice de transferencia múltiple y continua de un rango de parámetros probados antes de usarse: (solo fase de implementación) Rango de prueba para índices continuos y transferencias múltiples son especificadas en diseño y existen en código de implementación.
3. Subíndice de control: (solo fases de diseño e implementación) controles para valores de subíndices legales son especificados en diseño y codificados durante implementación.
4. Control razonable de parámetros de salida críticos durante el procesamiento: (solo fases de diseño e implementación) ciertos controles de un rango de valores son hechos durante el procesamiento para asegurar la sensatez de las salidas finales. esto es usualmente hecho solo para parámetros críticos. Estos son identificados durante diseño y codificados durante implementación

Fórmula: A nivel de sistema:

$$X=A/B,$$

A: Puntaje total de los elementos aplicables,

B: Número de elementos aplicables.

ELEMENTOS:

- (1): Seguir la definición,
- (2),(3) y (4): $Y=(1 - C/D)$,
C (Conteo): Número de módulos con violación,
D (Conteo): Número total de módulos

Atributo al que ayuda a calcular: Tolerancia al error

Influencia sobre otros atributos:

Granularidad: sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica en disciplinas de análisis, diseño, Implementación en el nivel del sistema

MÉTRICA 260

Nombre en español: Recuperación desde la lista de chequeo de errores desde HW

Nombre en inglés: Recovery from hardware fault checklist

Descripción: La métrica es la suma de los puntajes dados por los siguientes elementos dividido por el número de elementos aplicables:

1. Una declaración definitiva de requerimientos para recuperar fallas de hardware: (Solo requerimientos) El manejo de fallas de hardware como fallas aritméticas, de poder, interrupciones de reloj, etc. , son especificados durante la fase de requerimientos,

2. Recuperación de fallas de hardware:(Solo en diseño e implementación)
La especificación de diseño y código para proveer la recuperación de las fallas de hardware identificadas en los requerimientos deberían existir en las fases de diseño e implementación respectivamente

Fórmula: A nivel de sistema:

$$X=A/B,$$

A: Puntaje total de los elementos aplicables,

B: Número de elementos aplicables

Atributo al que ayuda a calcular: Tolerancia al error

Granularidad: Sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica en disciplinas de análisis, diseño, Implementación en el nivel del sistema

MÉTRICA 261

Nombre en español: Recuperación desde la lista de chequeo de errores de dispositivo

Nombre en inglés: Recovery from device errors checklist

Descripción: La métrica es la suma de los puntajes dados por los siguientes elementos dividido por el número de elementos aplicables:

1. Una declaración definitiva de requerimientos para recuperación de fallas de dispositivo: (Solo requerimientos) El manejo de fallas de dispositivo como condiciones inesperadas de fin-de-fichero o fin-de-grabación o fallas de lectura/escritura son especificados durante la fase de requerimientos,
2. Recuperación de fallas de dispositivo: (Solo en diseño e implementación) La especificación de diseño y código para proveer el manejo requerido de fallas de dispositivo deberían existir en las fases de diseño e implementación respectivamente

Fórmula: A nivel de sistema:

$$X=A/B,$$

A: Puntaje total de los elementos aplicables,

B: Número de elementos aplicables

Atributo al que ayuda a calcular: Tolerancia al error

Granularidad: Sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica en disciplinas de análisis, diseño, Implementación en el nivel del sistema

MÉTRICA 262

Nombre en español: Complejidad estructural

Nombre en inglés: Structural complexity

Descripción: Es una medida de control de la Complejidad estructural del software. Es el número de partes linealmente independientes cuando el software es ejecutado.

Fórmula:

$$S(i) = f^{out(i)},$$

$f^{out(i)}$: es la expansión del modulo i .

Atributo al que ayuda a calcular: Complejidad

Granularidad: método

Referencias: Choque [89], GONZÁLEZ [68], PRESSMAN [1]

Espectro: Gris

Otros: métrica de diseño arquitectónico, es posible que sea igual a "Cyclomatic complexity" (Métrica 269)

MÉTRICA 263

Nombre en español: Complejidad de datos

Nombre en inglés: Data complexity

Fórmula:

$$D(i) = v(i) / (f^{out(i)} + 1),$$

$f^{out(i)}$: es la expansión del modulo i ,

$v(i)$: es el número de variables de entrada y salida que entran y salen del modulo i .

Atributo al que ayuda a calcular: Complejidad

Referencias: Choque [89], GONZÁLEZ [68], PRESSMAN [1]

Espectro: Gris

Otros: es una métrica de diseño arquitectónico

MÉTRICA 264

Nombre en español: Complejidad de sistema

Nombre en inglés: Sistema complexity

Descripción: Suma de las Complejidades estructural y de datos. A medida que crecen los valores de Complejidad, la

Complejidad arquitectónica o global del sistema también aumenta. Esto lleva a una mayor probabilidad de que aumente el esfuerzo necesario para la integración y las pruebas

Fórmula:

$$C(i) = S(i) + D(i)$$

Atributo al que ayuda a calcular: Complejidad

Influencia sobre otros atributos:

Granularidad:

Referencias: Choque [89], GONZÁLEZ [68], PRESSMAN [1]

Espectro: Gris

Otros: es una métrica de diseño arquitectónico, Compuesta por las métricas 262 y 263

MÉTRICA 265

Nombre en español: Métrica de diseño arquitectónico

Nombre en inglés:

Descripción:

Fórmula:

$$MHK = \text{longitud}(i) * [\text{fin}(i) + \text{fout}(i)]^2,$$

Longitud (i): número de sentencias en lenguaje de programación en el modulo i y

fin (i) es la concentración del modulo i,

fout (i): es la expansión del modulo i

Atributo al que ayuda a calcular: Complejidad

Granularidad: Modulo

Referencias: GONZÁLEZ [68], PRESSMAN [1]

Espectro: Gris

Otros: es una métrica de diseño arquitectónico

MÉTRICA 266

Nombre en español: Respuesta para una clase

Nombre en inglés: Response for a class- RFC

Descripción: La métrica es el cardinal del conjunto de todos los métodos que pueden invocarse como respuesta a un mensaje a un objeto de la clase o como respuesta a algún método en la clase. Esto incluye a todos los métodos accesibles dentro de la jerarquía de la clase. Es decir, RFC es el número de métodos locales a una clase más el número de métodos llamados por los métodos locales. Es una medida de la Complejidad de una clase a través del número de métodos y de su comunicación con otras, pues incluye los métodos llamados desde fuera de la clase. Cuanto mayor es RFC, más Complejidad tiene el sistema, ya que es posible invocar más métodos como respuesta a un mensaje, exigiendo mayor nivel de comprensión, lo que implica mayor tiempo y esfuerzo de prueba y depuración

Fórmula:

$$RFC = |RS|, \text{ donde}$$

RS: es la respuesta de la clase, puede ser expresado como:

$$RS = \{M\} \cup \{R_i\}, \text{ donde}$$

{R_i}: conjunto de métodos llamados por el método i,

{M}: conjunto de todos los métodos en la clase.

Otra Forma:

$$RFC = A + B, \text{ donde}$$

A: Número de métodos locales,

B: Número de métodos llamados por métodos locales.

Atributo al que ayuda a calcular: Complejidad

Influencia sobre otros atributos: Acoplamiento, Comunicación, Analizabilidad

Granularidad: Clase, programa, sistema

Rango: Rango desde 0 hasta N, n es un número entero positivo

Referencias: VAZQUEZ [61], Chidamber y Kemerer [67], Gonzalez [68], PRESSMAN [1], OLMEDILLA [53], Lindell [86], MANSO [85]- Manso, Basili [70], El-Wakil [71], Li [72], Ojha [78], Lindroos [73], Andersson[74], Bar [75], Kan [24], LAIRD [76], Fenton [77], Garzas [57], Marin [28]

Espectro: Gris

Otros: Aplica en la disciplina de diseño. Problema: la definición es ambigua, Relacionada con DAC; MPC, (métrica 28, 29)

MÉTRICA 267

Nombre en español: Métodos ponderados por clase- MPC

Nombre en inglés: Weighted methods per class - WMC

Descripción: describe la Complejidad algorítmica de una clase en términos de las Complejidades de todos sus métodos. Está ligada a la calidad de la definición de Complejidad de un método (ci). Simplifica asignando 1 a cada método, convirtiéndose así en un simple contador del número de métodos dentro de una clase. En este caso habría que considerarla como una medida del tamaño de una clase y no de Complejidad, ya que una clase puede tener pocos métodos pero muy complejos y otra clase puede tener muchos métodos pero muy simples. Puede servir como un indicador de que una clase determinada necesite una descomposición adicional en varias clases

Fórmula:

Dada una clase C1 con los métodos M1,..., Mn definidos en la clase y c1,..., cn la Complejidad de los métodos y n es el número de métodos en una clase, WMC es definido como el sumatorio de las Complejidades de cada método de una clase.

$$WMC = \sum \text{desde } i=1 \text{ hasta } n \text{ de } [ci].$$

Si todos los métodos son considerados de igual Complejidad, entonces

$c1=1$ y $WMC=n$ (número de métodos).

Puede verse de otra forma

$WMC = \sum$ de todos los métodos $[wi]$, donde

wi : Peso estático asignado al método i de una clase

Atributo al que ayuda a calcular: Complejidad

Influencia sobre otros atributos: Mantenibilidad, Analizabilidad

Granularidad: Clase, programa, sistema

Referencias: VAZQUEZ [61], Chidamber y Kemerer [67], PRESSMAN [1], A17-Bansiya [31], OLMEDILLA [53], Lindell [86], Basili [70], El-Wakil [71], Li [72], Ojha [78], Lindroos [73], ANDERSSON[74], Bar [75], Kan [24], Laird [76], Fenton [77], Garzas [57], Marin [28]

Espectro: Claro

Otros: Aplica en la disciplina de diseño; es comparada con la métrica "Number of methods- NOM" (métrica 273), en este caso asume todas las Complejidades igual a 1. Algunas métricas consideradas como de peso son: Cyclomatic Complexity, LOC.

MÉTRICA 268

Nombre en español: Complejidad ciclomatica media

Nombre en inglés: Average v (G) or cyclomatic complexity

Descripción: Es la Complejidad ciclomatica media de los métodos de una clase. Es ofrecido como una medida de la Complejidad estructural. Una medida que esta concentrada en los métodos más complejos es la Complejidad ciclomática máxima (Maximum v(G)). Es un indicador útil de la dificultad de prueba y Mantenimiento de un programa. Sugiere un valor máximo de 10. La Complejidad ciclomática v(G) de un método es el número de caminos independientes a lo largo del método.

Fórmula:

En métodos Estructurados, $v(G) = \text{número de nodos de decisión} + 1$.

Otra formula esta dada por

$v(G) = e - n + 2$, donde,

e: número de aristas,

n: Número de nodos en el grafico.

Atributo al que ayuda a calcular: Complejidad

Granularidad: Clase, programa, sistema

Referencias: VAZQUEZ [61], McCabe [92], Mills [93]

Espectro: Gris

Otros: Aplica en la disciplina de implementación, métrica de la arquitectura si usa la formula

MÉTRICA 269

Nombre en español: Complejidad ciclomatica

Nombre en inglés: Cyclomatic Complexity v (G)

Descripción: propuesta inicialmente como una medida de una Complejidad lógica módulos de (McCabe [92]). El propósito principal de la métrica es evaluar la prueba y Mantenimiento de módulos de software, por lo que ha sido ampliamente utilizado en las áreas de investigación relacionadas con el Mantenimiento. En la práctica, la métrica es utilizada a menudo para calcular un límite inferior en el número de pruebas que deben ser diseñados y ejecutados para garantizar la cobertura de todas las declaraciones del programa en un módulo de software. Otro uso práctico de la métrica es que puede ser usado como un indicador de la fiabilidad de un sistema de software. Los estudios experimentales indican una fuerte correlación entre la métrica de McCabe [92] y el número de errores existentes en el código fuente, así como el tiempo necesario para encontrar y corregir tales errores. Puede ser un buen indicador de la Complejidad de un método, lo que indirectamente pueden utilizar para calcular la Complejidad de una clase. El máximo aceptado es 10

Fórmula:

$V(G) = e - n + p, O$, ó

$V(G) = e - n + 2, O$, ó

$V(G) = P + 1$, donde

p: es el número de nodos predicado (Un nodo que tiene más de una decisión), donde En el grafico de nodos G, con n vértices, e aristas, y p componentes conectados

Atributo al que ayuda a calcular: Complejidad

Influencia sobre otros atributos: Confiabilidad

Referencias: McCabe [92], ANDERSSON [74], McCabe [94], Lanza [88], Kan [24], LAIRD [76].

Espectro: Gris

Otros: También conocida como "Cyclomatic Number - CYCLO"

MÉTRICA 270

Nombre en español: Máxima Complejidad ciclomatica

Nombre en inglés: Max Cyclomatic Complexity

Descripción: Esta medida es la Complejidad ciclomatica máxima de las Complejidades individuales de las operaciones localmente definidas para la clase. Las Complejidades de las operaciones individuales son importantes y deberían mantenerse en un mínimo como en Funcionalidad del software desarrollado. La Complejidad máxima es un indicador del peor caso de Complejidad para la clase, y cuando es desarrollada mantiene la clase como un todo. Debe ser considerado cuando la calidad de la clase debe evaluarse.

Atributo al que ayuda a calcular: Complejidad

Referencias: PRITCHETT [69]

Espectro: Gris

Otros: Relacionada con la métrica 269

MÉTRICA 271

Nombre en español: Extensión a Complejidad ciclomatica

Nombre en inglés: Extensions to v (G) - Extensions cyclomatic complexity

Fórmula:

$$v'(G) = [l:u], \text{ Donde}$$

l y u: son saltos bajos y altos respectivamente

Atributo al que ayuda a calcular: Complejidad

Referencias: Mills [93]

Espectro: Gris

Otros: relacionada con la métrica 269. Puede que sea similar a métrica 272

MÉTRICA 272

Nombre en español: Métrica de Complejidad ciclomatica extendida

Nombre en inglés: Extended Cyclomatic Complexity Metric - ECC

Descripción: La métrica mide ECC lo mismo que el CC métricas, pero también tiene en cuenta las decisiones y bucles compuestos (Y / O)

Fórmula:

$$ECC = eV(G) = Pe + 1. \text{ Donde}$$

Pe: número de nodos predicados (Un nodo que tiene más de una decisión) en el grafico de flujo, ponderada por (añadir uno por cada declaración) el número de declaraciones compuestos (Y / O)

Atributo al que ayuda a calcular: Complejidad

Granularidad: modulo

Referencias: ANDERSSON [74]

Espectro: Gris

Otros: Puede que sea la misma métrica 271

MÉTRICA 273

Nombre en español: Número de métodos (locales)- NM

Nombre en inglés: Number of Methods (NOM) - Number of local methods

Descripción: Esta MÉTRICA es una cuenta de todos los métodos definidos en una clase. Podría concluirse que a mas métodos contenga una clase, es más alta la Complejidad de esa clase

Fórmula:

NOM= Número de métodos locales

Atributo al que ayuda a calcular: Complejidad

Granularidad: método, clase

Referencias: Bansiya [31], Lanza [88], LANGE [100], JETTER [3], OLMEDILLA [53], Lindell [86], Bellin [95], El-Wakil [71], Li [72], Bar [75], Garzas [57]

Espectro: Claro

Otros: Puede ser remplazada por "Weighted Methods per Class- WMC" (métrica 267), También llamada "Number of methds per class" (Número de métodos por clase)

MÉTRICA 274

Nombre en español: Número de clases

Nombre en inglés: Number of classes -NOC

Descripción: esta MÉTRICA puede dar una idea de toda la Complejidad de un sistema. Puede ser importante para determinar, después de que datos suficientes son recolectados desde diversos proyectos completados, una relación entre la Complejidad del dominio del problema y el número de clases

Fórmula:

NOC=Cuenta el número de clases en un sistema, no cuenta clases librería

Atributo al que ayuda a calcular: Complejidad, Tamaño

Granularidad: Clase

Referencias: Bellin [95], Lanza [88]

Espectro: Gris

MÉTRICA 275

Nombre en español: Número de paquetes

Nombre en inglés: Number of Packages- NOP

Descripción: El número de mecanismos de empaque de alto nivel

Fórmula:

NOP= número de paquetes

Atributo al que ayuda a calcular: Complejidad

Influencia sobre otros atributos: tamaño

Granularidad: sistema

Referencias: Lanza [88]

Espectro: Gris

MÉTRICA 276

Nombre en español: Flujo de información de Complejidad

Nombre en inglés: Information Flow Complexity -IFC

Fórmula:

$$IFC = (\text{fanin} * \text{fanout})^2,$$

fanin: número de flujos locales más el número de estructuras de datos que son utilizados como entrada,

fanout: es el número de flujos locales de salida de un módulo más el número de estructuras de datos que son utilizados como salida.

Atributo al que ayuda a calcular: Complejidad

Referencias: LAIRD [76]

Espectro: Gris

MÉTRICA 277

Nombre en español: Uso continuo de datos -externo

Nombre en inglés: Continued use of data

Descripción: ¿Puede el usuario o mantenedor continuar fácilmente usando los mismos datos después de reemplazar el software por uno anterior? ¿Ésta la migración del sistema software marchando exitosamente? Observe el comportamiento del usuario o mantenedor cuando el reemplaza el software por uno anterior.

Fórmula:

$$X = A/B,$$

A (Conteo): Número de datos que son usados en otro software a ser reemplazado y confirma que los datos están disponibles para ser usados continuamente,

B (Conteo): Número de datos que son usados en otro software a ser reemplazado y planeado a ser continuamente reusable

Atributo al que ayuda a calcular: Remplazabilidad

Rango: $0 \leq X \leq 1$, entre más grande mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 278

Nombre en español: Inclusión de funciones -externo

Nombre en inglés: Function Inclusiveness

Descripción: ¿Puede el usuario o mantenedor continuar fácilmente usando funciones similares después de reemplazar el software por uno anterior? ¿Ésta

la migración del sistema software marchando exitosamente? Observe el comportamiento del usuario o mantenedor cuando está remplazando el software por uno anterior

Fórmula:

$$X=A/B,$$

A (Conteo): Número de funciones que producen resultados similares a los producidos anteriormente y donde los cambios no han sido requeridos,

B (Conteo): Número de funciones probadas que son similares a funciones dadas por otro software a ser remplazado,

Atributo al que ayuda a calcular: Remplazabilidad

Rango: $0 \leq X \leq 1$, entre más grande mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

Otros: Personas involucradas: Desarrollador, Mantenedor, Operador

MÉTRICA 279

Nombre en español: Consistencia funcional de soporte de usuario

Nombre en inglés: User support functional consistency

Descripción: ¿Qué tan consistentes son los nuevos componentes con la interface de usuario existente? Observe el comportamiento del usuario y pregunte su opinion

Fórmula:

$$X=1- A1/A2,$$

A1 (Conteo): Número de nuevas funciones que el usuario encuentra inaceptablemente inconsistentes con la expectativa de usuario,

A2 (Conteo): Número de nuevas funciones,

Atributo al que ayuda a calcular: Remplazabilidad

Rango: $0 \leq X$, entre más grande es mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 280

Nombre en español: Uso continuo de datos -interno

Nombre en inglés: Continued use of data

Descripción: ¿Cuál es la cantidad de datos originales que permanecen sin cambios después de la sustitución de este producto? Cuenta el número de elementos de datos, que continúan usandose después de la sustitución tal como es especificado, y lo compara al número de elemento de datos requeridos para ser usados apartar de los datos antiguos después de la sustitución el software

Fórmula:

$$X=A/B,$$

A (Conteo): Número de elementos de datos que continúan siendo usados tal como la especificación después de la sustitución, confirmados en revisión,

B (Conteo): Número de elementos de datos antiguos que requieren ser usados apartar del software anterior, $0 \leq X \leq 1$, entre más grande mejor

Atributo al que ayuda a calcular: Remplazabilidad

Rango: $0 \leq X \leq 1$, entre más grande mejor

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 281

Nombre en español: Inclusión de funciones - interno

Nombre en inglés: Function Inclusiveness

Descripción: ¿Cuál es la cantidad de funciones que no cambian? Cuenta el número de funciones cubiertas por nuevos software que producen resultados similares y lo compara al número de funciones en el software anterior

Fórmula:

$$X=A/B,$$

A (Conteo): Número funciones cubiertas por el nuevo software que produce resultados similares, confirmados en revisión,

B (Conteo): Número de funciones en el software anterior

Atributo al que ayuda a calcular: Remplazabilidad

Rango: $0 \leq X \leq 1$, entre más grande mejor

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 282

Nombre en español: Tamaño de clase interfaz

Nombre en inglés: Class interface size (CIS)

Descripción: Esta MÉTRICA es una cuenta del número de métodos públicos en una clase

Fórmula:

$$\text{CIS} = \text{Número de métodos públicos de una clase}$$

Atributo al que ayuda a calcular: Mensajes

Granularidad: método-clase

Rango: 0 hasta N, donde N es entero positivo

Referencias: Bansiya [31], JETTER [3], OLMEDILLA [53], Garzas [57]

Espectro: Gris

Otros: Puede ser remplazada por la MÉTRICA número de métodos (MÉTRICA 273)

MÉTRICA 283

Nombre en español: Tamaño de diseño en clases

Nombre en inglés: Design size in classes (DSC) = Number of classes

Descripción: Es una cuenta del número total de clases en el diseño

Fórmula:

$$\text{DSC} = \text{Número de clases}$$

Número Total de clases en el diseño

Atributo al que ayuda a calcular: Tamaño

Granularidad: clase

Referencias: Bansiya [31], JETTER [3], OLMEDILLA [53], Garzas [57]

Espectro: Gris

Otros: Puede ser remplazada por la métrica "Number of classes" (Métrica 274). Es una métrica para aplicar en disciplina de diseño

MÉTRICA 284

Nombre en español: Size 1

Nombre en inglés: SIZE 1

Descripción: Es una variación de la tradicional LOC (Líneas de Código) definida específicamente para el lenguaje Ada

Fórmula:

$$\text{LOC} = \text{Número de punto y coma en la clase}$$

Atributo al que ayuda a calcular: Tamaño

Influencia sobre otros atributos: Analizabilidad

Referencias: OLMEDILLA [53], Lindell [86], Li [72], Bar [75], Garzas [57], Marin [28]

Espectro: Claro

Otros: La MÉTRICA puede llamarse Size 1 o LOC, Aplica en disciplina de implementación

MÉTRICA 285

Nombre en español: Size 2

Nombre en inglés: SIZE 2

Descripción: Número de atributos y métodos, simplemente cuenta el número de los atributos y los métodos de una clase. trata de una métrica orientada a objetos que puede ser aplicada a los lenguajes modulares considerando el número de variables (globalmente visible en un módulo) y su número de funciones y procedimientos

Fórmula:

$$\text{SIZE2} = \text{número de atributos} + \text{número de métodos locales.}$$

Atributo al que ayuda a calcular: Tamaño

Influencia sobre otros atributos: Analizabilidad, más probable que pertenezca a Complejidad

Granularidad: Clase

Referencias: OLMEDILLA [53], Lindell [86], Li [72], Bar [75], Garzas [57]

Espectro: Claro

Otros: También llamada "Número de atributos y métodos". Aplica en disciplina de diseño.

MÉTRICA 286

Nombre en español: Accesibilidad 1

Nombre en inglés: Accessibility 1

Descripción: ¿Están disponibles opciones al usuario tal que puedan ser consultadas características computacionales seleccionadas o salidas?

Atributo al que ayuda a calcular: Accesibilidad

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 287

Nombre en español: Accesibilidad 2

Nombre en inglés: Accessibility 2

Descripción: ¿El programa evita el uso de expresiones aritméticas habiendo incrustado constantes literales que están sujetas a cambio (e.g., CIRCUM = 3.14 * DIAM)?

Atributo al que ayuda a calcular: Accesibilidad

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 288

Nombre en español: Accesibilidad 3

Nombre en inglés: Accessibility 3

Descripción: ¿El programa permite para modificar la utilización de recursos, e.g., a través del uso de arrays de dimensión variable?

Atributo al que ayuda a calcular: Accesibilidad

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 289

Nombre en español: Accesibilidad 4

Nombre en inglés: Accessibility 4

Descripción: ¿El programa permite para variables exactitud computacional?

Atributo al que ayuda a calcular: Accesibilidad

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 290

Nombre en español: Accesibilidad 5

Nombre en inglés: Accessibility 5

Descripción: ¿El programa provee para omisión de variables innecesarias, cálculos y salidas para modos de generación opcionales?

Atributo al que ayuda a calcular: Accesibilidad

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 291

Nombre en español: Estructuración 1

Nombre en inglés: Structuredness 1

Descripción: ¿Han sido establecidas y seguidas reglas para transferencia de control entre módulos?

Atributo al que ayuda a calcular: Estructurado

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 292

Nombre en español: Estructuración 2

Nombre en inglés: Structuredness 2

Descripción: ¿Están limitados los módulos en tamaño?

Atributo al que ayuda a calcular: Estructurado

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 293

Nombre en español: Estructuración 3

Nombre en inglés: Structuredness 3

Descripción: ¿Ha sido seguido el orden: comentarios del bloque de cabecera, instancias de especificación y luego código ejecutable?

Atributo al que ayuda a calcular: Estructurado

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 294

Nombre en español: Estructuración 4

Nombre en inglés: Structuredness 4

Descripción: ¿Todos los subprogramas contienen, como máximo, un punto de salida?

Atributo al que ayuda a calcular: Estructurado

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 295

Nombre en español: Estructuración 5

Nombre en inglés: Structuredness 5

Descripción: ¿Tienen todos los subprogramas y funciones solamente un punto de entrada?

Atributo al que ayuda a calcular: Estructurado

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 296

Nombre en español: Estructuración 6

Nombre en inglés: Structuredness 6

Descripción: ¿El flujo del programa siempre avanza, con excepciones comentadas?

Atributo al que ayuda a calcular: Estructurado

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 297

Nombre en español: Estructuración 7

Nombre en inglés: Structuredness 7

Descripción: ¿La estructura exterior es consistente con la sub-secuencia de programas?

Atributo al que ayuda a calcular: Estructurado

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 298

Nombre en español: Estructuración 8

Nombre en inglés: Structuredness 8

Descripción: ¿El programa está subdividido en módulos de acuerdo con el reconocimiento sin esfuerzo de funciones?

Atributo al que ayuda a calcular: Estructurado

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 299

Nombre en español: Estructuración 9

Nombre en inglés: Structuredness 9

Descripción: ¿El programa esta escrito en un conjunto estándar de constructos disponibles en el lenguaje de programación particular usado (preferiblemente un estándar nacional como ANSI o IBM)?

Atributo al que ayuda a calcular: Estructurado

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 300

Nombre en español: Auto descripción 1

Nombre en inglés: Self- descriptness 1

Descripción: ¿Cada módulo[1] del programa contiene un bloque cabecera con comentarios que describe: (1) Nombre del programa; (2) fecha efectiva; (3) requisitos exactos; (4) propósito; (5) Limitaciones y restricciones; (6) Histórico de modificaciones; (7) Entradas y salidas; (8) Métodos; (9) Supuestos;

(10) Procedimiento para recuperación de errores para TODOS los errores previsibles que existen?

Atributo al que ayuda a calcular: Auto descripción

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 301

Nombre en español: Auto descripción 2

Nombre en inglés: Self- descriptness 2

Descripción: ¿Están descritos adecuadamente los puntos de decisión y consecuentes ramas de alternativas?

Atributo al que ayuda a calcular: Auto descripción

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 302

Nombre en español: Auto descripción 3

Nombre en inglés: Self- descriptness 3

Descripción: ¿Están adecuadamente definidas las funciones de los módulos tan bien como las entradas/salidas para permitir las pruebas de los módulos?

Atributo al que ayuda a calcular: Auto descripción

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 303

Nombre en español: Auto descripción 4

Nombre en inglés: Self- descriptness 4

Descripción: ¿Provee comentarios para soporte de selección de valores específicos que permita el desarrollo especializado de un programa de pruebas?

Atributo al que ayuda a calcular: Auto descripción

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 304

Nombre en español: Auto descripción 5

Nombre en inglés: Self- descriptness 5

Descripción: ¿Provee información para soportar la evaluación del impacto de un cambio en otra parte del programa?

Atributo al que ayuda a calcular: Auto descripción

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 305

Nombre en español: Auto descripción 6

Nombre en inglés: Self- descriptness 6

Descripción: ¿Provee información que soporte la identificación de código de programa donde debe ser modificado para efectos de un cambio requerido?

Atributo al que ayuda a calcular: Auto descripción

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 306

Nombre en español: Auto descripción 7

Nombre en inglés: Self- descriptness 7

Descripción: Donde hay dependencia de módulos, ¿está claramente especificado por comentarios, documentación de programa o estructura inherente al programa?

Atributo al que ayuda a calcular: Auto descripción

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 307

Nombre en español: Auto descripción 8

Nombre en inglés: Self- descriptness 8

Descripción: ¿Son representados los nombres descriptivos de las variables de la propiedad física o funcional?

Atributo al que ayuda a calcular: Auto descripción

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 308

Nombre en español: Auto descripción 9

Nombre en inglés: Self- descriptness 9

Descripción: ¿Únicamente funciones reconocibles contienen información descriptiva adecuada (e.g., comentarios) tal que el propósito de cada uno es claro?

Atributo al que ayuda a calcular: Auto descripción

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 309

Nombre en español: Auto descripción 10

Nombre en inglés: Self- descriptness 10

Descripción: ¿Provee de descriptores adecuados que permitan correlación de nombres de variables con la propiedad física o entidad que ella representa?

Atributo al que ayuda a calcular: Auto descripción

Referencias: Boehm [17]
Espectro: Oscuro

MÉTRICA 310

Nombre en español: Auto descripción 11
Nombre en inglés: Self- descriptness 11
Descripción: ¿Está explicado adecuadamente el código sin desarrollar?
Atributo al que ayuda a calcular: Auto descripción
Referencias: Boehm [17]
Espectro: Oscuro

MÉTRICA 311

Nombre en español: Cantidad de comentarios
Nombre en inglés: Quantity of comments
Descripción: La métrica el número de líneas comentadas dividido por el número total de líneas en cada modulo. Las líneas en blanco no son contadas. El valor promedio es calculado por la MÉTRICA a nivel de sistema
Fórmula:

$$X=A/B,$$

$$A=\sum ai,$$

ai: Cantidad de medidas de comentarios para cada modulo,

B: Número total de módulos,

Para cada modulo:

$$Y=C/D,$$

C: Número de comentarios (no en blanco)

D: número total de líneas (no en blanco)

Atributo al que ayuda a calcular: Auto descripción

Influencia sobre otros atributos:

Granularidad: Modulo, sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica en disciplina de implementación primero a nivel de modulo y después a nivel del sistema

MÉTRICA 312

Nombre en español: Efectividad de la medida de comentarios
Nombre en inglés: Effectiveness of comments measure
Descripción: La métrica es la suma de los puntajes de los siguientes elementos aplicables, dividido en número de elementos aplicables:
1. Módulos que tienen formato estándar en comentarios de prólogo: Los elementos que deben figurar en los comentarios de prólogo: Nombre del modulo/ Número de versión, autor, fecha, propósito entradas, salidas, función, suposiciones, limitaciones y restricciones,

requerimientos de exactitud, procedimientos de recuperación de error, referencias. Esta información es extremadamente valiosa para el personal nuevo que tienen que trabajar después con el software de desarrollado, la formación de pe-Mantenimiento, pruebas, cambios, etc., la medida a nivel del sistema esta basada en el número de módulos que no cumplen con un formato estándar o no proporcionar información completa.

2. Comentarios de código de manera uniforme: Líneas en blanco, bordes con asteriscos, tarjeta de columnas específicas, son algunas de las técnicas utilizadas para ayudar en la identificación de los comentarios. La medida esta basada en el número de módulos que no siguen las convenciones establecidas para activar los comentarios
3. Todas las transferencias de control y destinos comentados: Esta forma de ayudas en comentario en la comprensión y habilidad de seguir la lógica del módulo. La medida esta basada en el número de módulos que no cumplan.
4. Todas las máquinas dependen del código comentado: Comentarios asociados a la máquina dependiente de código son importantes no sólo para explicar lo que está haciendose, sino también sirve para identificar la parte del módulo dependiente de la máquina. La métrica esta basada en el número de módulos que no tienen código de la máquina comentado.
5. Todas las declaraciones no-estándar HOL comentadas: Una explicación similar a (4) es aplicable aquí.
6. Atributos de todas las variables declaradas comentadas: El uso, propiedades, unidades, etc., de variables aue explican en los comentarios. La medida esta basada en el número de módulos que no siguen esta práctica.
7. Comentarios no sólo repiten la operación descrita en el lenguaje: Los comentarios son para describir el ¿por qué no?. Un comentario, incrementa A de a 1, para la declaración $A=A + 1$ no proporciona nueva información. Un comentario, el incremento del cuadro de búsqueda de índice, es más valioso para la comprensión de la lógica del módulo. La medida esta basada en el número de módulos en los que las observaciones no explican el por qué.

Fórmula: A nivel de sistema:

$$X=A/B,$$

A: total de puntajes desde los elementos aplicables,

B: Número de elementos aplicables,

ELEMENTOS:

- (1), (2), (3), (4), (5), (6) Y (7): $Y=1-(C/D)$,
C: Número de módulos que violan la regla
D: número total de módulos

Atributo al que ayuda a calcular: Auto descripción

Granularidad: sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica a disciplina de implementación en el nivel del sistema

MÉTRICA 313

Nombre en español: Descripción de la medida de lenguaje de implementación

Nombre en inglés: Descriptiveness of implementación language measure

Descripción: La métrica es la suma de los puntajes de los siguientes elementos aplicables, dividido en número de elementos aplicables:

1. Lenguaje de alto nivel (HOL) utilizado: Un HOL es mucho más auto-descriptivo que el lenguaje ensamblador. La medida esta basada en el número de módulos que son implementados, completo o en parte, en ensamblador o lenguaje de máquina.
2. Formato estándar para organización de los módulos seguidos: Un formato específico ordenado como comentarios de prologo, expresiones declarativas, expresiones ejecutables son utilizados de manera uniforme en los módulos. Esta medida esta basada en el número de módulos que no cumplan con el formato estándar establecido.
3. Nombres de variables (mnemotécnicos) descriptivos de propiedades físicas o funcionales representadas: Mientras que la MÉTRICA parece muy subjetivo, es bastante fácil de identificar si los nombres de variables elegidas con mentalidad auto-descriptiva. Tres nombres de variables como el NAME, POSIT, SALRY son mucho mejores y más fácilmente reconocidos que A1, A2, A3. La medida esta basada en el número de módulos que no utilizan nombres descriptivos.
4. Código fuente lógicamente bloqueada e indentado: Técnicas como bloqueo, párrafos, indentación para construcciones específicas están bien establecidos y siguen de manera uniforme dentro de un sistema. Esta medida esta basada en el número de módulos que no cumplen con una técnica uniforme.
5. Una declaración por línea: El uso de declaraciones continuas y declaraciones múltiples por línea causa dificultad en la lectura del código. La medida es el número de líneas de declaraciones continuas más el número de líneas de múltiples declaraciones dividido por el

número total de líneas para cada módulo y después promediar sobre la totalidad de los módulos en el sistema.

6. Palabras del lenguaje no utilizadas como nombres: Algunos lenguajes permiten palabras clave para ser usados como etiquetas de declaración o como variables. Esta práctica es confuso para un lector. La medida esta basada en el número de módulos en los que utiliza una palabra clave de esta manera.

Fórmula: A nivel de sistema:

$$X=A/B,$$

A: total de puntajes desde los elementos aplicables,

B: Número de elementos aplicables,

ELEMENTOS:

- (1): $Y=1-(C/D)$,
C: Número de módulos con código directo.
D: número total de módulos;
- (2), (3), (4) Y (6): $Y=1-(C/D)$,
C: Número de módulos que violan la regla
D: número total de módulos;
- (5): $Y=1- (C+D)/E$,
C: Número líneas con declaraciones continuas,
D: número líneas con declaraciones múltiples,
E: Número total de líneas

Atributo al que ayuda a calcular: Auto descripción

Granularidad: sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica a disciplina de implementación en el nivel del sistema

MÉTRICA 314

Nombre en español: Requisitos de Desempeño asignados a diseño

Nombre en inglés: Performance requirements allocated to design (requirements)

Descripción: Requerimientos de Desempeño para el sistema deben ser desglosados y debidamente asignados a los módulos durante el diseño. Esta métrica identifica simplemente si tienen los requisitos de rendimiento (1) o no (0) fueron asignados durante el diseño

Fórmula: Seguir la definición

Atributo al que ayuda a calcular: Eficiencia de ejecución

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica a disciplina de implementación en el nivel del sistema

MÉTRICA 315

Nombre en español: Proceso iterativo de medida de Eficiencia

Nombre en inglés: Iterative processing efficiency measure (por modulo)

Descripción: La métrica en el nivel de modulo es la suma de los puntajes de los siguientes elementos aplicables dividido en número de elementos aplicables. A nivel de sistema es el puntaje promedio de todos los módulos:

1. Cálculos no dependientes del bucle manteniéndose fuera del bucle: Prácticas tales como la evaluación de las constantes en un bucle son evitadas. Esta medida esta basada en el número de declaraciones que no dependen del bucle encontradas en todos los bucles en un módulo. Esto es para ser medido a partir de una representación del diseño detallado durante el diseño y del código durante la implementación.
2. Optimización del rendimiento del lenguaje compilador / ensamblador utilizado: (sólo implementación) Esta es una medida binaria que identifica si un compilador de optimización de rendimiento ha sido utilizado 1, o si fue utilizado el lenguaje ensamblador para lograr la optimización de rendimiento 1, o no 0.
3. Expresiones compuestas definidas una vez: (Solo implementación) Reiteradas expresiones compuestas son evitadas desde el punto de vista de la Eficiencia. Esta MÉTRICA esta basada en el número de expresiones compuestas que aparecen más de una vez.
4. Número de capas: El uso de capas requiere más tiempo de procesamiento. Esta medida, la inversa del número de capas, refleja ese gasto de tiempo. Puede aplicarse durante el diseño, cuando el esquema de capas es definido y durante la implementación.
5. Libre empaquetado / desempaqueado de bits / byte en bucles: Esta es una medida binaria que indica el costo involucrados empaquetado / desempaqueado de bits / byte. La colocación de estas actividades dentro de los bucles debe evitarse si es posible. .
6. Libre de código ejecutable no funcional: (Solo en implementación) segmentos de código ejecutable que no realizan una función relevante son evidentes e ineficientes. Están presentes con mayor frecuencia durante la edición o rediseño al realizar actualizaciones sin la retirada completa de código obsoleto. Este elemento puede medirse sólo en la implementación y esta basada en el número de líneas de código ejecutable no funcionales aun.
7. Expresiones de decisión eficazmente codificada: (solo implementación) Esta medida esta basada en el número de expresiones de código de decisión ineficiente dividido por el número total de expresiones de decisión. Un ejemplo de expresión de código de decisión ineficiente no

tiene la alternativa de ejercicio más frecuentemente de la clausula SI expresión ENTONCES.

8. Vínculos de modulo: (solo implementación, requiere ejecución) Esta medida representa esencialmente la comunicación general entre el módulo. La medida esta basada en la cantidad de tiempo de ejecución gastado durante la comunicación módulo a módulo.
9. Vínculos del sistema operativo:(solo implementación, requiere ejecución) Esta medida representa el módulo de comunicación general del OS. La medida esta basada en la cantidad de tiempo de ejecución gastado durante la comunicación módulo a OS.

Fórmula: A nivel de modulo:

$$X=A/B,$$

A: total de puntajes desde los elementos aplicables,
B: Número de elementos aplicables;

Fórmula: A nivel de Sistema:

$$X=A/B,$$

A: $\sum a_i$, a_i : medidas de procesamiento iterativo para cada modulo,
B: Número total de módulos,

ELEMENTOS:

- (1): $Y=1-(C/D)$,
C: Número de declaraciones no dependientes de bucle en bucles.
D: número total de declaraciones de bucle;
- (2) y (5): seguir la definición
- (3) $Y=1-(C/D)$,
C: Número de expresiones compuestas definida más de una vez.
D: número expresiones compuestas;
- (4) $Y= 1 / D$, D: número capas;
- (6): $Y=1-(C/D)$,
C: Número de codigo ejecutable no funcional,
D: número total de declaraciones ejecutables;
- (7): $Y = (1- C/D)$,
C: Número declaraciones de decisión ineficientes,
D: número total de declaraciones de decisión;
- (8): $Y = (1- C/D)$,
C: tiempo de vinculo de modulo,
D: Tiempo de ejecución;
- (9): $Y = (1- C/D)$,
C: tiempo de vinculo de OS,
D: Tiempo de ejecución

Atributo al que ayuda a calcular: Eficiencia de ejecución

Influencia sobre otros atributos:

Granularidad: modulo, sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica a disciplinas de diseño e implementación

MÉTRICA 316

Nombre en español: Medida de Eficiencia de uso de datos

Nombre en inglés: Data usage efficiency measure (Por modulo)

Descripción: La métrica en el nivel de modulo es la suma de los puntajes de los siguientes elementos aplicables dividido en número de elementos aplicables. La métrica de sistema es el valor promedio de todos las métricas de los módulos:

1. Datos agrupados para procesamiento eficiente: Los datos utilizados por cualquier módulo son organizados en la base de datos, búferes, matrices, etc., de manera que facilite el procesamiento eficiente. La organización de datos durante el diseño y la implementación es examinada para dar esta medida binaria.
2. Variables inicializadas en el momento de la declaración: (Solo implementación) Esta medida esta basada en el número de variables que son utilizados en un módulo que no son inicializadas cuando son declaradas. Eficiencia es perdida cuando las variables son inicializadas durante la ejecución de una función o repetidamente inicializadas durante procesamiento iterativo.
3. Expresiones no mezcladas: Procesamiento general es consumido por el modo de mezcla de las expresiones que son innecesarios. Esta medida esta basada en el número de mezcla de expresiones encontradas en un módulo.
4. Elección común de unidades/tipos: Por motivos similares, tal como es expresado en el numeral 3 (anterior) esta convención debe seguirse. La medida es la inversa del número de operaciones realizadas que tengan unidades o tipos de datos no comunes.
5. Datos indexados o referenciados por procesamiento eficaz: No sólo la organización de datos, (1) anterior, pero el esquema de vínculos entre los datos efectúan el procesamiento eficientemente. Esta es una medida binaria de si la indexación utilizada para los datos ha sido seleccionada para facilitar el procesamiento.

Fórmula: A nivel de modulo:

$$X=A/B,$$

A: total de puntajes desde los elementos aplicables,

B: Número de elementos aplicables;

Fórmula: A nivel de Sistema:

$$X=A/B,$$

A: $\sum a_i$, a_i : medidas de uso de datos para cada elemento,

B: Número total de módulos,

ELEMENTOS:

- (1) y (5): seguir la definición
- (2): $Y=(C/D)$,
C: Número de inicializados cuando son declarados,
D: número total de variables;
- (3) $Y=1-(C/D)$,
C: Número de expresiones mezcla.
D: número expresiones ejecutables;
- (4) $Y= 1 / D$,
D: número de ocurrencias de operaciones de unidad no comunes

Atributo al que ayuda a calcular: Eficiencia de ejecución

Granularidad: modulo, sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica a disciplinas de diseño e implementación en el primer nivel de modulo

MÉTRICA 317

Nombre en español: Eficiencia de dispositivo 1

Nombre en inglés: Device efficiency 1

Descripción: ¿Han sido optimizados aquellos subprogramas que son ejecutados frecuentemente para aceleración de la ejecución?

Atributo al que ayuda a calcular: Eficiencia de dispositivo

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 318

Nombre en español: Eficiencia de dispositivo 2

Nombre en inglés: Device efficiency 2

Descripción: ¿Es usada la notación entera para todos los números de los exponentes?

Atributo al que ayuda a calcular: Eficiencia de dispositivo

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 319

Nombre en español: Eficiencia de dispositivo 3

Nombre en inglés: Device efficiency 3

Descripción: ¿Son usados repetidamente bloques COMMON de entidades temporales?

Atributo al que ayuda a calcular: Eficiencia de dispositivo

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 320

Nombre en español: Eficiencia de dispositivo 4

Nombre en inglés: Device efficiency 4

Descripción: ¿Son ingresadas variables por medio de sentencias de DATOS o BLOQUES DE DATOS?

Atributo al que ayuda a calcular: Eficiencia de dispositivo

Referencias: Boehm [17]

Espectro: Oscuro

MÉTRICA 321

Nombre en español: Lista de chequeo de comunicaciones común

Nombre en inglés: Communications commonality checklist

Descripción: La métrica es la suma de los puntajes de los siguientes elementos aplicables dividido por el número de elementos aplicables:

1. Declaración definitiva de los requerimientos de comunicación con otros sistemas: (requisitos) Durante la fase de requisitos, los requisitos de comunicación con otros sistemas deben ser considerados. Esta es una medida binaria de la existencia de esta consideración.
2. Estándares de Protocolo establecidas y seguidas: El estándar protocolo de comunicación para la comunicación con otros sistemas es establecido durante la fase de diseño y seguido durante la implementación. Esta es una medida binaria aplicada en cada una de estas fases, indica si los estándares han sido establecidos y seguidos.
3. Módulo de interfaz único para la entrada de otro sistema: a más módulos que manejan la entrada es más difícil hacer la interfaz con otro sistema y aplicar protocolos estándar. Esta medida esta basada en la inversa del número de módulos que manejan entradas aplicado a las especificaciones de diseño y código fuente.
4. Módulo de interfaz único para la salida a otro sistema: Por razones similares a (3) esta medida es la inversa del número de módulos de salida.

Fórmula: A nivel de sistema:

$$X=A/B,$$

A= puntaje total de los elementos aplicables,

B (conteo): Número de elementos aplicables.

ELEMENTOS:

- (1)y (2): Seguir la definición;

- (3)Y=1 /D,

D: Número de módulos usados para entrada

- 4): $Y=1 /D$,

D: Número de módulos usados para salida

Atributo al que ayuda a calcular: Estandarización de las comunicaciones

Granularidad: sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica a disciplinas de diseño e implementación a nivel de sistema

MÉTRICA 322

Nombre en español: Lista de chequeo de datos común

Nombre en inglés: Data commonality checklist

Descripción: La métrica es la suma de los puntajes de los siguientes elementos aplicables dividido por el número de elementos aplicables:

1. Declaración definitiva para la representación de datos estándar para comunicaciones con otros sistemas: (requisitos) Esta es una medida binaria de la existencia de la consideración para la representación de datos estándar entre los sistemas que han de interactuar. Esto debe ser abordado y medido en la fase de requisitos.
2. Estándar de traducción entre las representaciones establecido y seguido: (diseño e implementación) Más de una traducción de la representación de datos estándar utilizada para la interacción con otros sistemas pueden existir dentro de un sistema. Estándares para estas traducciones han sido establecidos y seguidos. Esta medida binaria identifica si los estándares son instituidos en el diseño y son seguidos durante la implementación.
3. Modulo único para realizar cada traducción: (diseño e implementación) Por razones similares a MÉTRICA 321. (3) y (4), esta medida es la inversa del número máximo de módulos que realizan una traducción

Fórmula: A nivel de sistema:

$$X=A/B,$$

A= puntaje total de los elementos aplicables, B (conteo): Número de elementos aplicables.

ELEMENTOS:

- (1) y (2): seguir a definición;
- (3): $Y= 1 /D$,

D: Número de módulos usados para realizar traducción

Atributo al que ayuda a calcular: Estandarización de los datos

Granularidad: sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica a disciplinas de diseño e implementación a nivel de sistema

MÉTRICA 323

Nombre en español: Facilidad de crecimiento 1

Nombre en inglés: Argmentability 1

Descripción: ¿Ha sido reservado algún núcleo de almacenamiento para soportar la extensión y proveer capacidades adicionales?

Atributo al que ayuda a calcular: Facilidad de crecimiento

Referencias: Boehm [17]

Espectro: oscuro

MÉTRICA 324

Nombre en español: Disponibilidad

Nombre en inglés: Availability

Descripción: ¿Cuan disponible es el sistema para uso durante un periodo especificado de tiempo? Sistema de prueba en una producción como entorno para un determinada periodo de tiempo realizando todas las operaciones de usuario. Medida del periodo de tiempo de reparación cada vez el sistema no está disponible durante el ensayo. Calcula el tiempo medio de reparación

Fórmula:

$$a) X = \{To/(To+Tr)\}$$

$$b) Y = A1/A2$$

To (Tiempo): Tiempo de operación,

Tr (Tiempo): Tiempo de reparación,

A1 (Conteo): total disponible en los casos de éxito de software del usuario cuando el usuario intenta utilizarlo

A2 (Conteo): número total de casos de intento del usuario para el uso del software durante el tiempo de observación. Esto es la función de llamado del usuario a la vista de operación

Atributo al que ayuda a calcular: Recuperabilidad

Rango: $0 \leq X \leq 1$ entre más largo y cerca a 1 es mejor, el usuario puede usar el software por más tiempo. $0 \leq Y \leq 1$, entre más largo y cerca a 1 es mejor

Referencias: ISO 9126 [15]:2, QUINT2 [58] [59]

Espectro: Gris

MÉTRICA 325

Nombre en español: Tiempo medio de inactividad

Nombre en inglés: Mean down time

Descripción: ¿Cuál es el tiempo promedio que el sistema permanece no disponible cuando una falla ocurre antes de ponerse en marcha gradual? Mide el tiempo de inactividad cada vez que el sistema no está disponible durante un periodo de ensayo especificado y calcula el tiempo medio.

Fórmula:

$$X=T/N,$$

T (Tiempo): Tiempo total de inactividad.

N (Cuenta): Número de descomposturas observadas. El peor caso o distribución de inactividad podría medirse.

Atributo al que ayuda a calcular: Recuperabilidad
Rango: $0 \leq X$, entre más pequeño mejor, el sistema caerá por menos tiempo
Referencias: ISO 9126 [15]:2
Espectro: Gris

MÉTRICA 326

Nombre en español: Reiniciabilidad
Nombre en inglés: Mean recovery time
Descripción: ¿Cuál es el tiempo promedio que el sistema toma para la recuperación completa desde la recuperación parcial inicial? Mide el tiempo de recuperación completa para cada vez que el sistema fue derribado durante el periodo de ensayo especificado y calcula el tiempo medio
Fórmula:

$$X = \sum (T)/N,$$

T (Tiempo): Tiempo de recuperación del sistema software derribado en cada oportunidad,

N (Conteo): Número de casos en los cuales el sistema software observado entra en recuperación.

Atributo al que ayuda a calcular: Recuperabilidad
Rango: $0 \leq X$, entre más pequeño mejor
Referencias: ISO 9126 [15]:2
Espectro: Gris

MÉTRICA 327

Nombre en español: Restaurabilidad -externo
Nombre en inglés: Restartability
Descripción: ¿Qué tan a menudo el sistema puede reiniciar prestar el servicio a usuarios dentro de un tiempo requerido? Cuenta el número de veces que el sistema reinicia y presta el servicio a usuarios dentro de un tiempo objetivo requerido y lo compara al número total de reinicios, cuando el sistema ha sido derribado durante un periodo de ensayo especificado
Fórmula:

$$X = A/B,$$

A (Conteo): Número de reinicios reunidos a tiempo necesarios durante las pruebas o soporte de operación de usuario,

B (Conteo): número total de reinicio durante las pruebas o las operaciones de soporte de usuario.

Atributo al que ayuda a calcular: Recuperabilidad
Rango: $0 \leq X \leq 1$, entre más grande y cerca a 1 es mejor, como el usuario puede reiniciar fácilmente.
Referencias: ISO 9126 [15]:2
Espectro: Gris

MÉTRICA 328

Nombre en español: Efectividad de restauración -externo

Nombre en inglés: Restore effectiveness

Descripción: ¿Qué tan efectiva es la capacidad de restauración? Cuenta el número de pruebas de restauración alcanzadas en tiempo de restauración y lo compara al número de restauraciones requeridas con un tiempo objetivo específico

Fórmula:

$$X=A/B,$$

A (Conteo): Número de casos de restauración exitosamente alcanzados en el tiempo de restauración objetivo,

B (Conteo): número de casos realizados .

Atributo al que ayuda a calcular: Recuperabilidad

Rango: $0 \leq X \leq 1$, entre más grande y cerca a 1 es mejor, como el proceso de restauración en el producto es mas

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 329

Nombre en español: Restaurabilidad -interno

Nombre en inglés: Restorability

Descripción: ¿Qué tan capaz es el producto de restaurarse a si mismo después de un evento anormal o en caso de solicitudes? Cuenta el número de requerimientos de restauración implementados y lo compara con el número de requerimientos de restauración en las especificaciones. Ejemplos de requerimientos de restauración: base de datos de control, punto de control de transacción, la función rehacer, función "deshacer"

Fórmula:

$$X=A/B,$$

A (Conteo): Número requerimientos de restauración implementados confirmados en revisión,

B (Conteo): número de requerimientos de restauración en los requerimientos.

Atributo al que ayuda a calcular: Recuperabilidad

Rango: $0 \leq X \leq 1$, donde X es grande mejor es la capacidad de restauración

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 330

Nombre en español: Efectividad de restauración -interno

Nombre en inglés: Restoration effectiveness

Descripción: ¿Qué tan efectiva es la capacidad de restauración? Cuenta el número de requerimientos de restauración reuniendo puntos de tiempo de restauración (mediante cálculos o simulaciones) y lo compara al número de requerimientos de restauración con tiempo objetivo específico

Fórmula:

$$X=A/B,$$

A (Conteo): Número de requerimientos de restauración implementados reuniendo puntos de tiempo de restauración,
B (Conteo): número de requerimientos de restauración con tiempos objetivo específico.

Atributo al que ayuda a calcular: Recuperabilidad

Rango: $0 \leq X \leq 1$, donde X es grande

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 331

Nombre en español: Tiempo medio entre fallos

Nombre en inglés: Mean time between failures (MTBF)

Descripción: ¿Cuan frecuentemente falla el software en operación? Cuenta el número de fallas ocurridas durante un periodo definido de operación y calcula el intervalo promedio entre las fallas

Fórmula:

$$a) X = T1/A$$

$$b) Y = T2/A$$

T1 (tiempo): tiempo de operación o período de tiempo total

T2 (tiempo): suma de intervalos de tiempo entre fallas consecutivas,

A(Conteo): Número total de fallas detectadas realmente (fallas ocurridas durante el tiempo de operación observado)

$$MTBF = MTTF + MTTR, \text{ donde,}$$

MTTF: tiempo medio de falla,

MTTR: tiempo medio en reparar

Atributo al que ayuda a calcular: Madurez

Rango: $0 \leq X, Y$, entre más largo es mejor. Como más tiempo puede ser esperado entre fallas.

Referencias: LAIRD [76], ISO 9126 [15]:2, Fitzpatrick [95]

Espectro: Gris

MÉTRICA 332

Nombre en español: Número de jerarquías

Nombre en inglés: Number of Hierarchies (NOH)

Descripción: Cuenta el número de clases jerárquicas en el diseño

Fórmula:

$$NOH = \text{número de clases con jerarquía}$$

Atributo al que ayuda a calcular: Jerarquía

Granularidad: Clase

Rango: desde 0 hasta N, N entero positivo

Referencias: Bansiya [31], JETTER [3], OLMEDILLA [53], Garzas [57]

Espectro: Gris

Otros: Puede ser remplazada por la métrica "Depth of Inheritance Tree" (Métrica 130)

MÉTRICA 333

Nombre en español: Densidad de falla latente estimada

Nombre en inglés: Estimated latent fault density

Descripción: ¿Cuántos problemas aún existen que pueden surgir como fallas futuras? Cuenta el número de fallas detectadas durante un periodo de ensayo definido y predice un número potencial de futuras fallas usando un modelo de estimación de crecimiento de fiabilidad

Fórmula:

$$X = \{ABS (A1-A2)\}/B$$

X: Estimado de densidad de falla latente residual

ABS (): Valor absoluto

A1 (Conteo): Número total de fallas latentes predichas en un producto software

A2 (Conteo): Número total de fallas detectadas realmente B (Tamaño): Tamaño del producto.

Atributo al que ayuda a calcular: Madurez

Rango: $0 \leq X$, depende del estado de las pruebas, en los escenarios grandes, más pequeño es mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 334

Nombre en español: Densidad de falla contra los casos de prueba

Nombre en inglés: Failure density against test cases

Descripción: ¿Cuántas fallas son detectadas durante el periodo de ensayo definido? Cuenta el número de fallos detectados casos de prueba realizados

Fórmula:

$$X = A1/A2$$

A1 (Conteo): Número de fallas detectadas

A2 (Conteo): Número casos de prueba realizados

Atributo al que ayuda a calcular: Madurez

Rango: $0 \leq X$, depende del estado de las pruebas, en los escenarios grandes, más pequeño es mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 335

Nombre en español: Resolución de fallas

Nombre en inglés: Failure Resolution

Descripción: ¿Cuántas condiciones de falla son resueltas? Cuenta el número de fallas que no ocurren durante el periodo de ensayo definido bajo condiciones similares. Mantiene el reporte de resolución de problemas describe el estado de las fallas.

Fórmula:

$$X=A1/A2$$

A1 (Conteo): Número de fallas resueltas

A2 (Conteo): Número total de fallas detectadas realmente

Atributo al que ayuda a calcular: Madurez

Rango: $0 \leq X \leq 1$, entre más cerca a 1 es mejor como mas fallas son resueltas

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 336

Nombre en español: Densidad de fallas

Nombre en inglés: Fault density

Descripción: ¿Cuántas fallas fueron detectadas durante el periodo de ensayo definido? Cuenta el número de fallas detectadas y densidad calculada

Fórmula:

$$X=A/B,$$

A (Conteo): Número de fallas detectado,

B (Tamaño): tamaño del producto.

Atributo al que ayuda a calcular: Madurez

Rango: $0 \leq X$ depende del estado de las pruebas, en los escenarios grandes, más pequeño es mejor

Referencias: ISO 9126 [15]:2, QUINT2 [58] [59]

Espectro: Gris

Otros: Tener cuidado de no contar fallas duplicadas

MÉTRICA 337

Nombre en español: Remoción de fallas- externo

Nombre en inglés: Fault removal

Descripción: ¿Cuántas fallas han sido corregidas? Cuenta el número de fallas removidas durante el proceso de pruebas y compara con el número total de fallas detectadas y el número de fallas predichas

Fórmula:

a) $X=A1/A2$

b) $Y=A1/A3$

A1 (Conteo): Número de fallas corregidas

A2 (Conteo): Número total de fallas detectadas realmente,

A3 (Conteo): Número total de fallas latentes predichas en el producto software

Atributo al que ayuda a calcular: Madurez

Rango: $0 \leq X \leq 1$, entre más cerca a 1 es mejor menos fallas permanecen

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 338

Nombre en español: Cobertura de prueba

Nombre en inglés: Test coverage (Specified operation scenario testing coverage)

Descripción: ¿Cuántos casos de prueba requeridos han sido ejecutados durante las pruebas? Cuenta el número de casos de prueba realizados durante las pruebas y compara el número de casos de prueba requeridos para obtener un adecuado alcance de pruebas

Fórmula:

$$X=A/B,$$

A (Conteo): Número de casos de prueba realizados actualmente representando un escenario de operación durante las pruebas,

B (Conteo): Número de casos de prueba para ser realizados para cubrir los requerimientos.

Atributo al que ayuda a calcular: Madurez

Influencia sobre otros atributos: Eficiencia

Rango: $0 \leq X \leq 1$ entre más cerca a 1 es mejor la cobertura de pruebas

Referencias: ISO 9126 [15]:2, QUINT2 [58] [59]

Espectro: Gris

MÉTRICA 339

Nombre en español: Madurez de prueba

Nombre en inglés: Test maturity

Descripción: ¿Está bien probado el producto? Nota: esto es para predecir la tasa de éxito que el producto alcanzara en futuras pruebas Cuenta el número de casos de prueba aprobados los cuales han sido realmente ejecutados y los compara al número total de casos de prueba a ser realizados conforme a los requerimientos

Fórmula:

$$X=A/B,$$

A (Conteo): Número de casos de prueba aprobados durante las pruebas o la operación,

B (Conteo): Número de casos de prueba para ser realizados para cubrir los requerimientos.

Atributo al que ayuda a calcular: Madurez

Rango: $0 \leq X \leq 1$ entre más cerca a 1 es mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 340

Nombre en español: Detección de falla

Nombre en inglés: Fault detection

Descripción: ¿Cuántas fallas fueron detectadas en la revisión del producto? Cuenta el número de fallas detectadas en revisión y lo compara con el número de fallas estimadas a ser detectadas en esta fase

Fórmula:

$$X=A/B,$$

A (Conteo): Número absoluto de fallas detectadas en revisión,

B (Conteo): número de fallas estimadas a ser detectadas en revisión (usando historia pasada o modelo de referencia),

Atributo al que ayuda a calcular: Madurez

Rango: $0 \leq X$ Un alto valor para X implica buena calidad del producto, mientras $A=0$ no necesariamente implica un estado libre de fallas de lo revisado.

Referencias: ISO 9126 [15]:3

Espectro: Gris

Otros: Esta métrica debería solo usarse para predicción durante el desarrollo

MÉTRICA 341

Nombre en español: Remoción de falla- interno

Nombre en inglés: Fault removal

Descripción: ¿Cuántas fallas han sido corregidas? ¿Cuál es la proporción de fallas removidas? Cuenta el número de fallas removidas durante diseño/codificación y lo compara al número de fallas detectadas en revisión durante diseño/codificación.

Fórmula:

$$X=A,$$

A (Conteo): Número de fallas corregidas en diseño/codificación,

$$Y=A/B,$$

A (Conteo): Número de fallas corregidas diseño/codificación,

B (Conteo): Número de fallas detectadas en revisión,

Atributo al que ayuda a calcular: Madurez

Rango: $0 \leq X$, Un alto valor de X implica, que quedan menos fallas, $0 \leq Y \leq 1$, entre más cerca esta Y de 1 es mejor (Mas fallas removidas)

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 342

Nombre en español: Adecuación de prueba

Nombre en inglés: Test adequacy

Descripción: ¿Cuanôtos de los casos de prueba requeridos son cubiertos por el plan de pruebas? Cuenta el número de casos de prueba planeados y lo compara al número de casos de prueba requeridos para obtener una adecuada cobertura de pruebas

Fórmula:

$$X=A/B,$$

A (Conteo): Número de casos de prueba diseñados en el plan de pruebas y confirmado en revisión,

B (Conteo): Número de casos de prueba requeridos

Atributo al que ayuda a calcular: Madurez

Rango: $.0 \leq X$, Donde X es grande esta la mejor suficiencia

Referencias: ISO 9126 [15]:3
Espectro: Gris

MÉTRICA 343

Nombre en español: Índice de madurez del software IMS

Descripción: Proporciona una indicación de la Estabilidad de un producto software basada en los cambios que ocurren con cada versión. A medida que el IMS sea cercano a 1.0 el producto empieza a estabilizar.

Fórmula:

$$\text{IMS} = [\text{Mt} - (\text{Fc} + \text{Fa} + \text{Fe})] / \text{Mt}, \text{ donde}$$

Mt: Número de módulos en la versión actual.

Fc: Número de módulos en la versión actual que han cambiado.

Fa: Número de módulos añadidos en la versión actual.

Fe: Número de módulos eliminados en la versión actual.

Atributo al que ayuda a calcular: Madurez

Influencia sobre otros atributos: Mantenibilidad

Referencias: CHOQUE [89], PRESSMAN [1]

Espectro: Gris

Otros: Aplica a disciplina de Mantenimiento

MÉTRICA 344

Nombre en español: Tiempo de respuesta - externo

Nombre en inglés: Response time

Descripción: ¿Cuál es el tiempo tomado para completar una tarea especificada? ¿Qué tanto toma antes la respuesta del sistema para una operación específica? Comienza una tarea específica. Mide el tiempo tomado para muestra de completar su operación. Mantiene un registro de cada intento

Fórmula:

$$T (\text{Tiempo}) = (\text{Tiempo de obtener el resultado}) - (\text{Tiempo de comando de entrada finalizado})$$

Atributo al que ayuda a calcular: Comportamiento en el tiempo

Rango: $0 < T$, entre más pronto mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 345

Nombre en español: Tiempo de respuesta (tiempo medio de respuesta)

Nombre en inglés: (a) Response time (Mean time to response)

Descripción: ¿Cuál es el tiempo de espera promedio de las experiencias del usuario después de emitir una solicitud antes de que la solicitud sea completada dentro de un sistema de carga específico en términos de tareas concurrentes y utilización del sistema? Ejecuta el número de escenarios de tareas concurrentes. Mide el tiempo que toma completar la(s) operación(es) seleccionadas. Mantiene el registro de cada intento y cálculo del tiempo medio para cada escenario

Fórmula:

$$X = T_{\text{mean}} / TX_{\text{mean}},$$

T_{mean} (Tiempo) = $\sum (T_i) / N$, (para $i=1$ hasta N)
 TX_{mean} (Tiempo) = Tiempo de respuesta medio requerido.
 T_i (Tiempo): tiempo de respuesta para la evaluación i -ésima (oportunidad),
 N (Conteo): número de evaluaciones (oportunidades de ejemplo).

Atributo al que ayuda a calcular: Comportamiento en el tiempo

Rango: $0 < X$, entre más cerca a 1 y menos que 1 es mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 346

Nombre en español: Tiempo de respuesta (peor caso en el cociente de tiempo de respuesta)

Nombre en inglés: Response time (Worst case response time ratio)

Descripción: ¿Cuál es el límite absoluto sobre el tiempo requerido para cumplir una función? ¿En el peor de los casos, puede aun el usuario obtener respuestas dentro del límite de tiempo especificado? ¿Ene el peor de los casos, puede aun el usuario obtener respuesta desde el software dentro de un corto tiempo suficiente a ser tolerable por el usuario? Calibrar la prueba. Simular una condición por la cual el sistema alcance una situación de carga máxima. Ejecutar la aplicación y monitorear resultados.

Fórmula:

$$X = T_{\text{max}} / R_{\text{max}},$$

T_{max} (Tiempo) = $\text{MAX} (T_i)$ (para $i=1$ hasta N),
 R_{max} (Tiempo) = tiempo de respuesta máximo requerido,
 $\text{MAX} (T_i)$: Tiempo máximo de respuesta entre evaluaciones,
 N (Conteo): Número de evaluaciones (Oportunidades de ejemplo),
 T_i (Tiempo): Tiempo de respuesta de la evaluación i -ésima (oportunidad).

Atributo al que ayuda a calcular: Comportamiento en el tiempo

Rango: $0 < X$, entre más cerca a 1 y menos que 1 mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 347

Nombre en español: Rendimiento

Nombre en inglés: Throughput

Descripción: ¿Cuántas tareas pueden ser realizadas exitosamente sobre un periodo de tiempo dado? Calibrar cada tarea a la prioridad pretendida dada. Comenzar diversas tareas de trabajo. Mide el tiempo que toma para la tarea medida para completar su operación. Mantiene un registro de cada intento.

Fórmula:

$$X = A / T,$$

A (Conteo): Número de tareas completadas,
T (Tiempo): Periodo de tiempo de observación.

Atributo al que ayuda a calcular: Comportamiento en el tiempo

Rango: $0 < X$, entre más largo es mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 348

Nombre en español: Rendimiento (Cantidad media de rendimiento)

Nombre en inglés: Throughput (Mean amount of throughput)

Descripción: ¿Cuál es el número promedio de tareas concurrentes que el sistema puede manejar sobre un conjunto de unidades de tiempo? Calibrar cada tarea a la prioridad pretendida dada. Ejecutar el número de tareas concurrentes. Mide el tiempo que toma completar la tarea seleccionada en el tráfico dado. Mantiene un registro de cada intento.

Fórmula:

$$X = X_{\text{mean}} / R_{\text{mean}},$$

$$X_{\text{mean}} (\text{Conteo}) = \sum (X_i) / N,$$

$R_{\text{mean}} (\text{Conteo}) =$ Rendimiento medio requerido.

$$X_i (\text{Tiempo}) = A_i / T_i,$$

$A_i (\text{Conteo}):$ Número de tareas concurrentes observadas sobre un periodo de tiempo para la evaluación i -ésima,

$T_i (\text{Tiempo}):$ Periodo de tiempo para la evaluación i -ésima ,

$N (\text{Conteo}):$ número de evaluaciones.

Atributo al que ayuda a calcular: Comportamiento en el tiempo

Rango: $0 \leq X$, entre más largo es mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 349

Nombre en español: Rendimiento (peor caso en el cociente de rendimiento)

Nombre en inglés: Throughput (Worst case throughput ratio)

Descripción: ¿Cuál es el límite absoluto sobre el sistema en términos del número y manejo de tareas concurrentes como rendimiento? Calibrar la prueba. Simular la condición por la cual el sistema alcanza una situación de carga máxima. Ejecutar tareas de trabajo concurrentemente y monitorear resultados.

Fórmula:

$$X = X_{\text{max}} / R_{\text{max}},$$

$$X_{\text{max}} (\text{Conteo}) = \text{MAX} (X_i) (\text{para } i=1 \text{ hasta } N)$$

$R_{\text{max}} (\text{Conteo}):$ Rendimiento máximo requerido,

$\text{MAX} (X_i):$ Número máximo de tareas de trabajo entre evaluaciones,

$$X_i (\text{Conteo}) = A_i / T_i,$$

$A_i (\text{Conteo}):$ Número de tareas concurrentes observadas sobre un periodo de tiempo de la evaluación i -ésima,

T_i (Tiempo): periodo de tiempo de la evaluación i-ésima,
N (Conteo): Número de evaluaciones.

Atributo al que ayuda a calcular: Comportamiento en el tiempo

Rango: 0 < X, entre más largo es mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 350

Nombre en español: Tiempo de espera - externo

Nombre en inglés: Turnaround time

Descripción: ¿Cuál es el tiempo de espera de las experiencias del usuario después de emitir una instrucción para comenzar un grupo de tareas relacionadas y su finalización? Calibrar el test consecuentemente. Comenzar las tareas de trabajo. Mide el tiempo que toma para las tareas de trabajo completar su operación. Mantiene un registro de cada intento.

Fórmula:

T (Tiempo)= Tiempo entre que el usuario final obtenga resultados de salida y la petición del usuario final.

Atributo al que ayuda a calcular: Comportamiento en el tiempo

Rango: 0 < T, entre más corto es mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

Otros: Es recomendable tomar en cuenta el ancho de banda del tiempo y usar análisis estático con medidas para muchas tareas, no solo una tarea

MÉTRICA 351

Nombre en español: Tiempo de espera (tiempo medio de espera)

Nombre en inglés: Turnaround time (Mean time for turnaround)

Descripción: ¿Cuál es promedio de tiempo de espera de las experiencias de usuario después de emitir una instrucción para comenzar un grupo de tareas relacionadas y su finalización dentro de la carga de sistema especificada en términos de tareas concurrentes y utilización de sistema? Calibrar la prueba. Simular una condición donde una carga es colocada sobre el sistema por la ejecución de una serie de tareas (oportunidades de ejemplo). Mide el tiempo que toma completar la tarea de trajo seleccionada en el trafico dado. Mantiene un registro de cada intento

Fórmula:

$X = T_{\text{mean}} / TX_{\text{mean}}$,

T_{mean} (Tiempo) = $\sum (T_i) / N$, (for $i=1$ to N)

TX_{mean} (Tiempo)=Tiempo de respuesta medio requerido.

T_i (Tiempo): tiempo de respuesta para la evaluación i-esima (oportunidad),

N (Conteo): número de evaluaciones (oportunidades de ejemplo).

Atributo al que ayuda a calcular: Comportamiento en el tiempo

Rango: 0 < X, entre más corto mejor.

Referencias: ISO 9126 [15]:2
Espectro: Gris

MÉTRICA 352

Nombre en español: Tiempo de espera (peor caso del cociente de tiempo de espera)

Nombre en inglés: Turnaround time (Worst case turnaround time ratio)

Descripción: ¿Cuál es el límite absoluto sobre el tiempo requerido en la realización de una tarea de trabajo? ¿En el peor de los casos, que tanto le toma al sistema software realizar tareas especificadas? Calibrar la prueba. Simular una condición donde el sistema alcance la carga máxima en términos de la tarea realizada. Ejecute la tarea de trabajo seleccionada y monitorear resultados.

Fórmula:

$$X = T_{\max} / R_{\max},$$

T_{\max} (Tiempo) = MAX (Ti) (for $i=1$ to N),
 R_{\max} (Tiempo) = tiempo de respuesta máximo requerido,
MAX (Ti): Tiempo máximo de respuesta entre evaluaciones,
N (Conteo): Número de evaluaciones (Oportunidades de ejemplo),
Ti (Tiempo): tiempo de respuesta para la evaluación i -ésima (oportunidad).

Atributo al que ayuda a calcular: Comportamiento en el tiempo

Rango: $0 < X$, Entre más cerca a 1 y menor a 1 es mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 353

Nombre en español: Tiempo esperando

Nombre en inglés: Waiting time

Descripción: ¿Qué proporción de tiempo gastan los usuarios esperando para que el sistema responda? Ejecuta el número de escenarios de tareas concurrentes. Mide el tiempo que toma para completar la operación(es) seleccionada. Mantiene un registro de cada intento y calcula el tiempo medio para cada escenario.

Fórmula:

$$X = T_a / T_b,$$

T_a (Tiempo): Tiempo total gastado esperando,
 T_b (Tiempo): tiempo de tarea.

Atributo al que ayuda a calcular: Comportamiento en el tiempo

Rango: $0 \leq X$, entre más corto mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 354

Nombre en español: Tiempo de respuesta - interno

Nombre en inglés: Response time

Descripción: ¿Cuál es el tiempo estimado para completar una tarea específica? Evalúa la Eficiencia del sistema operativo y la aplicación de las llamadas al sistema. Estima el tiempo de respuesta basados en esto. Lo siguiente puede ser medido: -la totalidad o parte de las especificaciones de diseño - prueba completa de camino de transacción -prueba completa de módulos / partes del producto de software -producto de software completo durante la fase de prueba

Fórmula:

$$X (\text{Tiempo}) = \text{time (calculado o simulado)},$$

Atributo al que ayuda a calcular: Comportamiento en el tiempo

Rango: entre más pequeño sea el cociente es mejor

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 355

Nombre en español: Tiempo de rendimiento – interno

Nombre en inglés: Throughput time

Descripción: ¿Cuál es el número estimado de tareas que pueden realizarse en más de una unidad de tiempo? Evalúa la Eficiencia del manejo de recursos en el sistema. Hacer un factor basado en la aplicación de llamadas al sistema en manejo de recursos

Fórmula:

$$X (\text{Conteo}) = \text{Ninguna de las tareas por unidad de tiempo.}$$

Atributo al que ayuda a calcular: Comportamiento en el tiempo

Rango: Entre más grande es el cociente mejor

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 356

Nombre en español: Tiempo de espera – interno

Nombre en inglés: Turnaround time

Descripción: ¿Cuál es el tiempo estimado para completar un grupo de tareas relacionadas con un puesto de trabajo como mucho? Evalúa la Eficiencia del sistema operativo y la aplicación de las llamadas al sistema. Estima el tiempo de respuesta para completar un grupo de tareas relacionadas basadas en esto. Lo siguiente puede ser medido: - la totalidad o parte de las especificaciones de diseño. - prueba completa de camino de transacción. - prueba completa de módulos / partes del producto de software. -producto de software completo durante la fase de pruebas

Fórmula:

$$X (\text{Tiempo}) = \text{Time (calculado o simulado).}$$

Atributo al que ayuda a calcular: Comportamiento en el tiempo

Rango: Entre más corto es el cociente mejor

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 357

Nombre en español: Capacidad de rastro de auditoría

Nombre en inglés: Audit trail capability

Descripción: ¿Puede el usuario identificar una operación específica que causa falla? ¿Puede el mantenedor encontrar fácilmente una operación específica que causa falla? Observar el comportamiento del usuario o mantenedor quien está tratando de resolver las fallas

Fórmula:

$$X=A/B,$$

A (Conteo): Número de datos realmente registrados durante la operación,

B (Conteo): Número de datos a registrar suficiente para monitorear la condición del software durante la operación.

Atributo al que ayuda a calcular: Analizabilidad

Rango: $0 \leq X$, entre más cerca a 1 mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 358

Nombre en español: Soporte de la función de diagnóstico

Nombre en inglés: Diagnostic function support

Descripción: ¿Qué tan capaces son las funciones de diagnóstico en los análisis causales de soporte? ¿Puede el usuario identificar la operación específica que causó falla? (El usuario puede ser capaz de evitar caer en la misma ocurrencia de falla nuevamente con la operación alternativa) ¿Puede el mantenedor encontrar fácilmente la causa de la falla? Observar el comportamiento del usuario o mantenedor quien está tratando de resolver los fallos usando funciones de diagnóstico

Fórmula:

$$X=A/B$$

A (Conteo): Número de fallas las cuales el mantenedor puede diagnosticar (Usando la función de diagnóstico) para entender la causa-efecto de la relación,

B (Conteo): Número total de fallas registradas

Atributo al que ayuda a calcular: Analizabilidad

Rango: $0 \leq X \leq 1$, entre más cerca a 1 mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 359

Nombre en español: Capacidad de análisis de falla

Nombre en inglés: Failure analysis capability

Descripción: ¿Puede el usuario identificar una operación específica que causa falla? ¿Puede el mantenedor encontrar fácilmente la falla? Observe el comportamiento del usuario o el mantenedor quien está tratando de resolver las fallas.

Fórmula:

$$X=1- A/B$$

A (Conteo): Número de fallas en las que las causas no son encontradas aún,

B (Conteo): Número total de fallas registradas

Atributo al que ayuda a calcular: Analizabilidad

Rango: $0 \leq X \leq 1$, entre más cerca a 1 mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 360

Nombre en español: Eficiencia de análisis de falla

Nombre en inglés: Failure analysis efficiency

Descripción: ¿Puede el usuario analizar eficientemente la causa de la falla? (Algunas veces el usuario realiza Mantenimiento de un conjunto de parámetros). ¿Puede el mantenedor encontrar fácilmente la causa de la falla? Observe el comportamiento del usuario o mantenedor quien está tratando de resolver fallas.

Fórmula:

$$X=\sum (T) /N,$$

T (Tiempo): Tout-Tin

Tout (Tiempo): Momento en que las causas de falla son encontradas (o reportadas de regreso por el usuario),

Tin (Tiempo): Momento en el que el reporte de la falla es recibido,

N (Conteo): Número de falas registradas

Atributo al que ayuda a calcular: Analizabilidad

Rango: $0 \leq X$, entre más corto es mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 361

Nombre en español: Capacidad de monitorear estado

Nombre en inglés: Status monitoring capability

Descripción: ¿Puede el usuario identificar una operación específica la cual causa falla por monitoreo de datos durante la operación?. ¿Puede el mantenedor encontrar fácilmente datos durante la operación? Observar el comportamiento de usuario o mantenedor quien está tratando de monitorear datos registrando estado del software durante la operación

Fórmula:

$$X=1- A/B$$

A (Conteo): Número de casos los en cuales el mantenedor (o usuario) falla en monitorear datos,

B (Conteo): Número de casos en los cuales el mantenedor (o usuario) intenta monitorear datos registrando estado del software durante la operación.

Atributo al que ayuda a calcular: Analizabilidad

Rango: $0 \leq X \leq 1$, entre más cerca a 1 mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 362

Nombre en español: Registro de actividad

Nombre en inglés: Activity recording

Descripción: ¿Cuan minucioso es el registro del estado del sistema? Cuenta el número de artículos registrados en el registro de actividad tal como es especificado y debe ser comparado al número de artículos que requieren ser registrados.

Fórmula:

$$X=A/B$$

A (Conteo): Número de elementos de registro de datos implementados tal como es especificado, confirmación en revisión,

B (Conteo): Número de elementos a ser registrados definidos en las especificaciones.

Atributo al que ayuda a calcular: Analizabilidad

Rango: $0 \leq X \leq 1$, entre más cerca a 1, más datos proveen sistema de registro

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 363

Nombre en español: Preparación de la función de diagnostico

Nombre en inglés: Readiness of diagnostic function

Descripción: ¿Cuan minuciosa es la provisión de funciones de diagnostico?

Fórmula:

$$X=A/B$$

A (Conteo): Número de funciones de diagnostico implementadas tal como es especificado, confirmación en revisión,

B (Conteo): Número de funciones de diagnostico requeridas.

Atributo al que ayuda a calcular: Analizabilidad

Rango: $0 \leq X$, entre más cerca a 1, la mejor implementación de funciones de diagnostico

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 364

Nombre en español: Violación de Complejidad

Nombre en inglés: Complexity violation

Descripción: Mide el número de casos de prueba de TTCN-3 [56], funciones y alt steps que violan un valor límite definido de una medida de la Complejidad en comparación con el número total de casos de prueba, funciones y altsteps. Varias medidas de la Complejidad pueden ser utilizadas, por ejemplo, Número ciclomático de McCabe [92], o el nivel de anidamiento

Fórmula:

$$X=1- A/B$$

A: número de entidades de comportamiento violando el límite superior de la Complejidad,

B: número total de entidades de comportamiento

Atributo al que ayuda a calcular: Analizabilidad

Referencias: Zeiss [56]

Espectro: Gris

MÉTRICA 365

Nombre en español: Porcentaje de éxito del cambio

Nombre en inglés: Change success ratio

Descripción: ¿Puede el usuario operar el sistema software sin fallas después del Mantenimiento? ¿Puede el mantenedor mitigar fácilmente las fallas causadas por efectos secundarios de Mantenimiento? Observe el comportamiento de usuario o mantenedor quien está operando el sistema software después del Mantenimiento. Cuenta las fallas las cuales el usuario o mantenedor encuentran durante la operación del software antes y después del Mantenimiento. Por otra parte, investiga el reporte de resolución de problemas, el reporte de operación o reporte de Mantenimiento.

Fórmula:

$$X=Na/Ta,$$

$$Y= \{(Na/Ta)/(Nb/Tb)\},$$

Na (Conteo): Número de casos en los cuales el usuario encontró fallas durante la operación después de que el software cambio,

Nb (Conteo): Número de casos en los cuales el usuario encontró fallas durante la operación antes de que el software cambio,

Ta (Tiempo): tiempo de operación durante el periodo de observación especificado después de que el software cambio

Tb (Tiempo): tiempo de operación durante el periodo de observación especificado antes de que el software cambio

Atributo al que ayuda a calcular: Estabilidad

Rango: $0 \leq X, Y$, entre más pequeño y ceca a cero es mejor.

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 366

Nombre en español: Modificaciones a la localización de impacto – externo

Nombre en inglés: Modification impact localisation (Emerging failure after change)

Descripción: ¿Puede el usuario operar el sistema software sin fallas después del Mantenimiento? ¿Puede el mantenedor mitigar fácilmente las fallas causadas por efectos secundarios de Mantenimiento? Cuenta las ocurrencias de fallas después del cambio, las cuales están mutuamente encadenadas y afectadas por el cambio.

Fórmula:

$$X=A/N,$$

A (Conteo): número de fallas surgidas después la falla resuelta por cambios durante un periodo especificado,

N (Conteo): número de fallas resueltas

Atributo al que ayuda a calcular: Estabilidad

Rango: $0 \leq X$, entre más pequeño y cerca a 0 es mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 367

Nombre en español: Impacto del cambio

Nombre en inglés: Change impact

Descripción: ¿Cuál es la frecuencia de impactos adversos después de la modificación? Cuenta el número de impactos adversos detectados después de la modificación y lo compara al número de modificaciones realizadas

Fórmula:

$$X=1- A/B$$

A (Conteo): Número de impactos adversos detectados después de las modificaciones,

B (Conteo): Número de modificaciones hechas.

Atributo al que ayuda a calcular: Estabilidad

Rango: $0 \leq X$, entre más cerca a 0 es mejor

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 368

Nombre en español: Modificaciones a la localización de impacto – interno

Nombre en inglés: Modification impact localization

Descripción: ¿Qué tan grande es el impacto de la modificación en el producto software? Cuenta el número de variables afectadas desde una modificación y lo compara al número total de variables en el producto

Fórmula:

$$X=A/B,$$

A (Conteo): Número de variables de datos de variable afectadas por modificación, confirmado en revisión,

B (Conteo): Número total de variables.

Atributo al que ayuda a calcular: Estabilidad

Rango: $0 \leq X \leq 1$, entre más cerca a 0, menor impacto de la modificación

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 369

Nombre en español: Variable global y uso de temporizador

Nombre en inglés: global variable and timer usage

Descripción: Las variables globales pueden promover los efectos secundarios. En TTCN-3 [56], las variables de los componentes y los temporizadores son globales a todos los comportamientos son ejecutadas en el mismo componente. Esta métrica mide, el número de todas las variables de los componentes y temporizadores de referencia en más de una función, caso de prueba, o altstep y los relaciona con el número total de las variables de componente y temporizadores

Fórmula:

$$X = 1 - A/B,$$

A: número de variables de los componentes y temporizadores de referencia para más de un comportamiento,

B: número de todas las variables de componentes y temporizadores

Atributo al que ayuda a calcular: Estabilidad

Referencias: Zeiss [56]

Espectro: Oscuro

MÉTRICA 370

Nombre en español: Reasignación de parámetro

Nombre en inglés: parameter reassignment

Descripción: Cualquier modificación de los parámetros que son pasados en un caso de prueba, función o altstep como parámetro de salida o entrada conduce a un efecto secundario. Por lo tanto, esta métrica mide el potencial de efectos secundarios mediante la relación del número de parámetros de salida y entrada con el número total de parámetros.

Fórmula:

$$X = 1 - A/B,$$

A: número de parámetros de salida y entrada,

B: número total de parámetros

Atributo al que ayuda a calcular: Estabilidad

Referencias: Zeiss [56]

Espectro: Gris

MÉTRICA 371

Nombre en español: Co existencia disponible - externo

Nombre en inglés: Available co-existence

Descripción: ¿Qué tan a menudo el usuario encuentra algunas restricciones o fallas inesperadas cuando está operando concurrentemente con otro

software? Usar el software evaluado concurrentemente con otros software que el usuario utiliza a menudo

Fórmula:

$$X=A/T,$$

A (Conteo): Número de cualquier restricción o falla inesperada que el usuario encuentra durante la operación concurrente con otro software,

T (Tiempo): tiempo de operación de concurrentemente operar otro software.

Atributo al que ayuda a calcular: Co existencia

Rango: $0 < X$, entre mas ceca a 0 es mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 372

Nombre en español: Co existencia disponible - interno

Nombre en inglés: Available co-existence

Descripción: ¿Qué tan flexible es el producto en compartir su entorno con otros productos sin efectos adversos sobre otros productos? Cuenta el número de entidades con las cuales el producto puede co-existir tal como en la especificación y lo compara al número de entidades en entorno de producción que requiere co-existencia

Fórmula:

$$X=A/B,$$

A (Conteo): Número de entidades con la cuales el producto puede co-existir tal como en la especificación,

B (Conteo): Número de entidades en entorno de producción que requieren co-existir.

Atributo al que ayuda a calcular: co existencia

Rango: $0 \leq X \leq 1$, entre más cerca a 1 mejor

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 373

Nombre en español: Eficiencia del ciclo de cambio

Nombre en inglés: Change cycle efficiency

Descripción: ¿Puede los problemas de usuario ser resueltos para su satisfacción dentro de una escala de tiempo aceptable? Monitor de interacción entre usuario y proveedor. Registra el tiempo tomado desde el primer requerimiento de usuario para la resolución del problema

Fórmula:

$$\text{Tiempo Promedio: } T_{av} = \sum(T_u)/N.$$

$$T_u = T_{rc} - T_{sn}$$

T_{sn} = tiempo en el cual el usuario finaliza de enviar requerimientos para Mantenimiento al proveedor con el reporte de problemas.

Trc= Tiempo en el cual el usuario recibe la liberación de la versión revisada (o reporte de estatus).

N= Número e visiones revisadas.

Atributo al que ayuda a calcular: Facilidad de cambio

Rango: $0 < T_{av}$, Entre más pequeño mejor, a excepción de que el número e versiones sea grande.

Referencias: ISO 9126 [15]:2

Espectro: Gris

Otros: Tu (Tiempo), Trc (Tiempo), Tsn (Tiempo), N (conteo), Tav (Tiempo)

MÉTRICA 374

Nombre en español: Tiempo transcurrido de implementación de cambio

Nombre en inglés: Change implementación elapsed time

Descripción: ¿Puede el mantenedor cambiar el software fácilmente para resolver problemas de fallas? Observación de comportamiento de usuario mantenedor mientras tratan de cambiar el software. Por otra parte, investiga el reporte de resolución de problemas o reporte de Mantenimiento

Fórmula:

$$\text{Tiempo Promedio: } T_{av} = \sum(T_m)/N.$$

$T_m = T_{out} - T_{in}$

T_{out} = tiempo en el cual las causas de fallos son removidas mediante la alteración del software.

T_{in} = Tiempo en el cual las causas de las fallas son encontradas. N= Número de fallas registradas y removidas.

Atributo al que ayuda a calcular: Facilidad de cambio

Rango: $0 < T_{av}$, Entre más pequeño mejor, a excepción de que el número e fallas sea grande

Referencias: ISO 9126 [15]:2

Espectro: Gris

Otros: Tout (Tiempo), Tin (Tiempo), Tm (Tiempo), N (conteo), Tav (Tiempo)

MÉTRICA 375

Nombre en español: Modificación de la Complejidad

Nombre en inglés: Modification complexity

Descripción: ¿Puede el mantenedor cambiar el software fácilmente para resolver el problema? Observación de comportamiento de mantenedor quien está tratando de cambiar el software. Por otra parte, investiga el reporte de resolución de problemas o reporte de Mantenimiento o descripción del producto.

Fórmula:

$$T = \sum(A/B)/N$$

A (tiempo): tiempo de abajo gastado en cambios,

B (tamaño): tamaño del software cambiado (puede ser cambiado por expresiones ejecutables de código de programa, número de ítems

cambiados de requerimientos, o paginas cambiadas del documento, etc.),

N (cuenta): número de cambios.

Atributo al que ayuda a calcular: Facilidad de cambio

Rango: $0 < T(\text{tiempo})$ entre más pequeño mejor o el número requerido de cambios es excesivo

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 376

Nombre en español: Modificabilidad de parámetros

Nombre en inglés: Parameterized modifiability

Descripción: ¿Puede el usuario o el mantenedor cambiar el parámetro fácilmente para cambiar el software y resolver los problemas? Observación de comportamiento del usuario o mantenedor mientras está tratando de cambiar el software. Por otra parte, investiga el reporte de resolución de problemas o reporte de mantenimiento.

Fórmula:

$$X = 1 - A/B,$$

A (Conteo): número de casos los que fallas de mantenimiento para cambiar el software por uso de parámetros,

B (Conteo): número de casos los cuales el mantenedor intenta cambiar el software usando parámetros.

Atributo al que ayuda a calcular: Facilidad de cambio

Rango: $0 \leq X \leq 1$, entre más cerca a 1 mejor

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 377

Nombre en español: Capacidad de control de cambios del software

Nombre en inglés: Software change control capability

Descripción: ¿Puede el usuario identificar fácilmente versiones revisadas?, ¿Puede el mantenedor cambiar el software para resolver problemas? Observación de comportamiento del usuario o mantenedor mientras está tratando de cambiar el software. Por otra parte, investiga el reporte de resolución de problemas o reporte de mantenimiento.

Fórmula:

$$X = A/B,$$

A (Conteo): número de cambios de datos log realmente actualizados,

B (Conteo): número de cambios datos log planeados para ser grabados lo suficiente para rastrear cambios en el software.

Atributo al que ayuda a calcular: Facilidad de cambio

Rango: $0 \leq X \leq 1$, entre más cerca a 1 mejor o entre más cerca a 0 menos tienen lugar a tomar

Referencias: ISO 9126 [15]:2

Espectro: Gris

MÉTRICA 378

Nombre en español: Capacidad de registro de cambios

Nombre en inglés: Change recordability

Descripción: ¿Son los cambios para especificaciones y módulos del programa adecuadamente grabados en el código con líneas de comentarios? Cociente de registro del modulo de información de cambios

Fórmula:

$$X=A/B,$$

A (Conteo): número de cambios en Funciones/módulos que tienen comentarios de cambios confirmados en revisión,

B (Conteo): número total de funciones/módulos cambiados desde el código original.

Atributo al que ayuda a calcular: Facilidad de cambio

Rango: $0 \leq X \leq 1$, entre más cerca a 1 mas grabable. El cambio de control 0 indica un pobre control o pocos cambios, alta Estabilidad

Referencias: ISO 9126 [15]:3

Espectro: Gris

MÉTRICA 379

Nombre en español: Duplicación de código

Nombre en inglés: Code duplication

Descripción: Dado que los cambios a código duplicado requieren la modificación de todos los lugares de la duplicación, este indicador determina la porción de código duplicado en términos de, por ejemplo, Líneas de código (LOC) o declaraciones

Fórmula:

$$X=1- A/B,$$

A: Entidades que contienen código duplicado,

B: número total de entidades

Atributo al que ayuda a calcular: Facilidad de cambio

Referencias: Zeiss [56]

Espectro: Oscuro

MÉTRICA 380

Nombre en español: Número máximo de violación de referencias

Nombre en inglés: maximum number of references violation

Descripción: Este indicador determina con qué frecuencia existe referencia a una entidad y penaliza la violación de un valor límite superior. Al aplicar los cambios a las entidades que refiere n muy a menudo, que un desarrollador necesita para comprobar todas las referencias de si un cambio puede tener efectos secundarios no deseados o requiere el seguimiento de los cambios.

Fórmula:

$$X=1- A/B,$$

A: Número de entidades que son referenciadas más veces que un límite superior permitido,

B: número total de entidades

Atributo al que ayuda a calcular: Facilidad de cambio

Referencias: Zeiss [56]

Espectro: Oscuro

MÉTRICA 381

Nombre en español: Punto de función - PF

Nombre en inglés: Function point

Descripción: medida para el tamaño. Es manejado de la siguiente forma:

Parámetro de ponderación	Factor de ponderación		
	Simple	Medio	Complejo
Número de entradas del usuario	3	4	6
Número de salidas del usuario	4	5	7
número de consultas del usuario	3	4	6
Número de archivos	7	10	15
Número de interfaces externas	5	7	10

El valor de la cuenta es dado para cada parámetro de medición, luego es multiplicado por el factor de ponderación correspondiente, y finalmente los resultados son sumandos para la cuenta total

Fórmula:

$$PF= \text{cuenta-total} \times (0,65 + 0,01 \times \sum(F_i))$$

Donde

cuenta-total: es la \sum a de todas las entradas PF obtenidas

F_i son «los valores de ajuste de Complejidad».

Para ello deben tenerse en cuenta: número de entradas del usuario, número de salidas del usuario, número de consultas de usuario, número de archivos, número de interfaces externas y ajustar el factor de ponderación para cada medida del modelo de análisis.

F_i : calificar cada una de las siguientes preguntas de 0 (no aplicable) a 5 (absolutamente esencial)

1. ¿Requiere el sistema copias de seguridad y de recuperación fiables?
2. ¿Requiere comunicación de datos?
3. ¿Existen funciones de procesamiento distribuido?
4. ¿Es crítico el rendimiento?
5. ¿Es el sistema ejecutado en un entorno operativo existente y fuertemente utilizado?

6. ¿Requiere el sistema entrada de datos interactiva?
7. ¿Requiere la entrada de datos interactiva que las transacciones de entrada sean llevadas a cabo sobre múltiples pantallas u operaciones?
8. ¿Son los archivos maestros actualizados de forma interactiva?
9. ¿Son complejas las entradas, las salidas, los archivos o las peticiones?
10. ¿Es complejo el procesamiento interno?
11. ¿Es el código diseñado para ser reutilizable?
12. ¿Están incluidas en el diseño la conversión y la conversión y la instalación'?
13. ¿Es el sistema diseñado para soportar múltiples instalaciones en diferentes organizaciones?
14. ¿Es la aplicación diseñada para facilitar los cambios y para ser fácilmente utilizada por el usuario?

Atributo al que ayuda a calcular: Tamaño

Influencia sobre otros atributos: Analizabilidad, Cambiabilidad

Referencias: CHOQUE [89], PRESSMAN [1], Mills [93], Kan [24], Marin [28]

Espectro: Claro

MÉTRICA 382

Nombre en español: Métrica del bang

Nombre en inglés: Bang Measure

Descripción: puede emplearse para desarrollar una indicación del tamaño del software a implementar como consecuencia del modelo de análisis. Para calcular la métrica bang, el desarrollador de software debe evaluar primero un conjunto de primitivas (elementos del modelo de análisis que no son subdivididos más en el nivel de análisis):

1. Primitivas funcionales (PFu).
2. Elementos de datos (ED).
3. Objetos (OB)
4. Relaciones (RE).
5. Estados (ES).
6. Transiciones (TR).

Cuentas adicionales:

- Primitivas modificadas de función manual (PMFu)
- Elementos de datos de entrada (EDE).
- Elementos de datos de salida (EDS).
- Elementos de datos retenidos (EDR).
- Muestras (*tokens*) de datos (TCi).
- Conexiones de relación (REi).

Atributo al que ayuda a calcular: tamaño
Referencias: PRESSMAN [1], MILLS [93]- Mills
Espectro: Gris

MÉTRICA 383

Nombre en español: Número de elementos

Nombre en inglés: Number of elements

Descripción: Define una consulta cuenta de todas las entidades que existen en el meta modelo. Da una indicación del tamaño del modelo. Además asigna pesos a todos los tipos de elementos de la arquitectura y ahí permite una vista sobre qué elementos son usados extensivamente y cuáles no o apenas son usados

Fórmula:

A nivel de casos de uso:

A=Número de casos de uso existente en el diagrama de casos de uso.

A nivel de escenario:

A=Número de clases en el modelo.

Atributo al que ayuda a calcular: Tamaño

Referencias: LANGE [100]

Espectro: Gris

MÉTRICA 384

Nombre en español: Líneas de código

Nombre en inglés: Lines of Code per método - LOC

Descripción: La métrica es el número de líneas activas de código (líneas ejecutables) en un método. El tamaño de un método es empleado para evaluar la Facilidad de comprensión, la capacidad de Reutilización y la Facilidad de Mantenimiento del código.

Depende del lenguaje de programación y de la Complejidad de los métodos. En sistemas OO el número de líneas de código de los métodos debería ser bajo. Lorenz y Kidd [87] sugieren un umbral de 24 LOC para métodos en C++ y de 8 en SmallTalk para descubrir qué métodos son candidatos a ser divididos en varios. No es una métrica recomendada para sistemas OO, pero es fácil de recoger y utilizar.

Fórmula:

LOC= Conteo de líneas de código

Atributo al que ayuda a calcular: Tamaño

Influencia sobre otros atributos: Complejidad

Granularidad: Método, clase, programa, sistema

Rango: desde 0 hasta N, donde N entero positivo

Referencias: VAZQUEZ [61], Lorenz y Kidd [87], Lanza [88], Kan [24], Mills [93], Li [72], ANDERSSON[74], Bar [75]

Espectro: Claro

Otros: Aplica en disciplina de implementación

MÉTRICA 385

Nombre en español: Métrica de tamaño

Fórmula:

Tamaño= $n + a$, donde

n: Número de nodos (módulos).

a: es el número de arcos (líneas de control)

Atributo al que ayuda a calcular: Tamaño

Referencias: PRESSMAN [1], Fenton [77]

Espectro: Gris

Otros: Aplica en la disciplina de diseño (arquitectura)

MÉTRICA 386

Nombre en español: Métrica de profundidad

Fórmula:

Profundidad= el camino más largo desde el nodo raíz (parte más alta) a un nodo hoja.

Atributo al que ayuda a calcular: Tamaño

Referencias: PRESSMAN [1], Fenton [77]

Espectro: Gris

Otros: esta relacionada con la Métrica Depth of inheritance tree (métrica 130).

Aplica en la disciplina de diseño (arquitectura)

MÉTRICA 387

Nombre en español: Métrica de anchura

Fórmula:

Anchura= máximo número de nodos de cualquier nivel de la arquitectura

Atributo al que ayuda a calcular: Tamaño

Referencias: PRESSMAN [1], Fenton [77]

Espectro: Gris

Otros: Aplica en la disciplina de diseño (arquitectura)

MÉTRICA 388

Nombre en español: Métrica de relación arco-nodo

Descripción: Mide la densidad de la conectividad de la arquitectura y puede proporcionar una sencilla indicación del acoplamiento de la arquitectura

Fórmula:

Relación arco-nodo= a/n , donde

n: Número de nodos (módulos).

a: es el número de arcos (líneas de control)

Atributo al que ayuda a calcular: tamaño

Influencia sobre otros atributos: acoplamiento

Referencias: PRESSMAN [1], Fenton [77]

Espectro: Gris

Otros: Aplica en la disciplina de diseño (arquitectura)

MÉTRICA 389

Nombre en español: Número de métodos públicos

Nombre en inglés: Number of Public methods - NPM

Descripción: Esta es una cuenta del número de métodos públicos en una clase. Es usada para ayudar a estimar la cantidad de trabajo a desarrollar de una clase

Fórmula:

Atributo al que ayuda a calcular: Tamaño

Granularidad: clase

Referencias: PRESSMAN [1], EL-WAKIL [71]

Espectro: Gris

Otros:

MÉTRICA 390

Nombre en español: Número de variables publicas por clase

Nombre en inglés: Number of public variables per class - NVP

Descripción: Esta Métrica cuenta el número de variables publicas en una clase. Lorenz y Kidd consideran el número de variables en una clase como una medida de su tamaño. El hecho de que una clase posee más variables públicas que otra, podría dar a entender que la clase tiene más relaciones con otros objetos, y como tal es más probable que sea una clase clave

Fórmula:

NPV= número de atributos públicos de la clase

Atributo al que ayuda a calcular: Tamaño

Granularidad: clase

Referencias: PRESSMAN [1], EL-WAKIL [71]

Espectro: Gris

MÉTRICA 391

Nombre en español: Número de variables por clase

Nombre en inglés: Number of variable per clase -NV

Descripción: El número total de variables incluidas variables públicas, privadas y protegidas.

Fórmula:

NV: Número total de atributos de la clase.

Atributo al que ayuda a calcular: Tamaño

Influencia sobre otros atributos: composición

Granularidad: clase

Referencias: PRESSMAN [1], EL-WAKIL [71]

Espectro: Gris

Otros: Puede ser la misma que métrica 44

MÉTRICA 392

Nombre en español: Número de métodos de clase

Nombre en inglés: Number of Class Methods - NCM

Descripción: Este indicador contabiliza el número total de métodos de la clase en una clase. Un método de clase es un método que es global para sus instancias

Fórmula:

NCM= número de total de métodos de la clase

Atributo al que ayuda a calcular: tamaño

Granularidad: clase

Referencias: PRESSMAN [1], EL-WAKIL [71], Genero [90]

Espectro: Gris

MÉTRICA 393

Nombre en español: Número de métodos instancia públicos

Nombre en inglés: Number of public instance method- PIM

Descripción: Esta Métrica contabiliza el número total de métodos de instancia pública en una clase. Los métodos públicos son los que están disponibles como servicios a otras clases.

Fórmula:

PIM= número de métodos públicos de la clase

Atributo al que ayuda a calcular: Tamaño

Granularidad: clase

Referencias: Genero [90]

Espectro: Gris

MÉTRICA 394

Nombre en español: Número de métodos instancia

Nombre en inglés: Number of instance methods - NIM

Descripción: Esta cuenta métricas todos los públicos, protegidos, y los métodos definidos para los casos particulares de clase

Fórmula:

NIM= número de métodos definidos en la clase

Atributo al que ayuda a calcular: tamaño

Granularidad: Clase

Referencias: Genero [90]

Espectro: Gris

Otros: Presenta similitud con la métrica 392

MÉTRICA 395

Nombre en español: Número de variables instancia

Nombre en inglés: Number of instance variables - NIV

Descripción: Este indicador contabiliza el número total de variables de instancia en una clase. Las variables de instancia son las variables privadas y protegidas a disposición de las instancias

Fórmula:

NIV= número de variables de la clase

Atributo al que ayuda a calcular: tamaño
Granularidad: clase
Referencias: Genero [90]
Espectro: Gris
Otros: Tiene similitud con la métrica 28

MÉTRICA 396

Nombre en español: Número de mensajes enviados
Nombre en inglés: Number of messages send- NOM
Descripción: NOM mide el número de mensajes enviados en un método, segregados por el tipo de mensaje. Los tipos incluyen: • Unarios: mensajes sin argumentos. • Binarios: mensajes con un argumento que pertenecen a tipos especiales (por ejemplo, concatenación y funciones matemáticas). • Clave: mensajes con uno o más argumentos. • Valoración: NOM cuantifica el tamaño del método de una manera relativamente no sesgada. Lorenz y Kidd sugieren un umbral de 9. Un valor alto puede indicar un estilo funcional y/o una colocación pobre de responsabilidades. Lenguajes como C++ pueden llamar a sistemas no OO, y estas llamadas no deberían contarse en el número de mensajes enviados.

Atributo al que ayuda a calcular: Tamaño
Granularidad: método
Referencias: Genero [90]
Espectro: Gris
Otros: Aplica a disciplina de implementación

MÉTRICA 397

Nombre en español: Tamaño relativo
Nombre en inglés: relative size
Descripción: es una métrica para estructuras multi diagrama UML
Fórmula:

- A) $RS = \text{Número de diagramas de secuencia} / \text{número de casos de uso}$
- B) $RS = \text{Número de objetos} / \text{número de clases.}$
- C) $RS = \text{Número de cuadros de estado} / \text{número de clases}$

Atributo al que ayuda a calcular: tamaño
Referencias: A40- Lange
Espectro: oscuro

MÉTRICA 398

Nombre en español: Índice de mantenibilidad
Nombre en inglés: Maintainability Index - MI
Descripción: Las funciones de MI son esencialmente compuestas de las siguientes tres medidas tradicionales: métrica de McCabe [92] Complejidad Ciclomática, el volumen de Halstead y la MÉTRICA LOC. En la práctica, MI métricas puede ser un buen indicador de que un sistema viejo puede estar en necesidad de re-ingeniería, o cuando un sistema en fase de desarrollo tiene

que ser re-diseñado, porque está tornandose imposible de mantener. Los valores umbral siguientes para MI en un módulo fueron propuestos (derivados de Fortran, C y código Ada):

- MI <65 => Mantenimiento deficiente
- 65 = <MI <85 => Mantenimiento justo
- MI > = 85 => Mantenimiento excelente

Fórmula:

La primera métrica para MI es denominada índice de Mantenibilidad no comentado

$$(NC): MI = 171 - 5.2 * \ln(\text{aveV}) - 0.23 * \text{aveV}(g') - 16.2 * \ln(\text{aveLOC}),$$

Donde:

aveV: es el volumen de Halstead por modulo,

aveV(g'): es el promedio de la Complejidad ciclomatica extendida por modulo,

aveLOC: es el promedio de líneas de código por modulo.

La segunda métrica para MI es denominada Índice de Mantenibilidad no comentado:

$$MI = 171 - 5.2 * \ln(\text{aveV}) - 0.23 * \text{aveV}(g') - 16.2 * \ln(\text{aveLOC}) + 50 * \sin(\sqrt{2.4 * \text{perCM}}),$$

Donde

aveV: es el volumen de Halstead por modulo,

aveV(g'): es el promedio de la Complejidad ciclomatica extendida por modulo,

aveLOC: es el promedio de líneas de código por modulo,

perCM: es el porcentaje promedio de comentario por modulo,

Atributo al que ayuda a calcular: Tamaño

Influencia sobre otros atributos: Mantenibilidad

Referencias: ANDERSSON [74], Laird [76]

Espectro: Gris

Otros: Hace uso de métricas como: "LOC" (métrica 384), "Cyclomatic complexity" (métrica 269) y "Extended Cyclomatic complexity" (métrica 272)

MÉTRICA 399

Nombre en español: Medida de Eficiencia de almacenamiento

Nombre en inglés: Storage efficiency measure (por modulo)

Descripción: La métrica en el nivel de modulo es la suma de los puntajes de los siguientes elementos aplicables dividido en número de elementos aplicables. a nivel de sistema es el valor promedio de todos los valores de los módulos:

1. Requisitos de almacenamiento asignado a diseño: (solo diseño) Los requerimientos de almacenamiento para el sistema son asignados a cada uno de los módulos durante el diseño. Esta medida es binaria, en el caso de "sí" es explícitamente hecha (1), en el caso de "no" es (0).

2. Facilidades de almacenamiento virtual usadas: El uso de almacenamiento virtual o las técnicas de paginación aumenta la Eficiencia de almacenamiento de un sistema. Esta es una medida binaria, las técnicas han previsto (1) para “sí” o (0) para “no”.
3. Datos comunes definidos solo una vez: (solo implementación) A menudo, los datos globales o los datos utilizados comúnmente son definidos más de una vez. Esto consume el almacenamiento. Esta medida esta basada en el número de variables que son definidas en un módulo que han sido definidas en otros lugares.
4. Programa de Segmentación: los esquemas de segmentación eficiente minimizan al máximo de la longitud del segmento a minimizar el requisito de almacenamiento. Esta medida esta basada en la máxima longitud del segmento. Esto Aplica durante el diseño, cuando están disponibles estimaciones y durante la implementación.
5. Datos de Segmentación: La cantidad de datos referenciados por un módulo en forma de matrices, los buffers de entrada, o de datos globales, a menudo es pequeño en comparación con el tamaño de las zonas de almacenamiento necesario. Esto representa un uso ineficiente de almacenamiento. La medida esta basada en la cantidad de datos no utilizados, dividido por la cantidad total de datos disponibles para un módulo.
6. administración de memoria dinámica usada: Esta es una medida binaria enfatizando las ventajas de usar técnicas de administración de memoria dinámica para minimizar el conteo de almacenamiento requerido durante ejecución. planeada durante diseño y usada durante implementación.
7. Empaquetamiento de datos usado: (implementación), mientras el empaquetamiento de datos fue desalentador en métrica 315, en circuitos debido a la sobrecarga que es añadida al tiempo de procesamiento, en general, es beneficioso desde el punto de vista de la Eficiencia de almacenamiento. Esta medida binaria es aplicada durante la ejecución reconociendo este hecho. .
8. Libertad o código no funcional: Código no funcional, ya sea ejecutable (véase métrica 315) o no, consume espacio de almacenamiento por lo que no es deseable. Esta medida esta basada en el número de líneas de código que no son funcionales. .
9. Código no duplicado: Código duplicado debe ser evitado por la misma razón como en (8). La medida aplicada durante el diseño e implementación esta basada en la cantidad de código duplicado

10. Lenguaje de Compilador/ensamblador de optimización de almacenamiento:(Implementación) Esta medida binaria es similar a métrica 315, excepto en el punto de vista de la optimización del almacenamiento.

11. Libre de elementos de datos redundantes:(Implementación), Esta medida refiere a la base de datos y esta basada en el número de elementos de datos redundantes.

Fórmula: A nivel de modulo:

$$X=A/B,$$

A: total de puntajes desde los elementos aplicables

B: Número de elementos aplicables;

Fórmula: A nivel de Sistema:

$$X=A/B,$$

A: $\sum a_i$, a_i : medidas de procesamiento iterativo para cada modulo,

B: Número total de módulos

ELEMENTOS:

- (1): $Y=1-(C/D)$,
C: Número de declaraciones no dependientes de bucle en bucles.
D: número total de declaraciones de bucle;
- (2) y (5): seguir la definición
- (3) $Y=1-(C/D)$,
C: Número de expresiones compuesta definida más de una vez.
D: número expresiones compuestas;
- (4) $Y= 1 / D$,
D: número capas; (
- 6): $Y=1-(C/D)$,
C: Número de código ejecutable no funcional,
D: número total de declaraciones ejecutables;
- (7): $Y = (1- C/D)$,
C: Número declaraciones de decisión ineficientes,
D: número total de declaraciones de decisión;
- (8): $Y = (1- C/D)$,
C: tiempo de vinculo de modulo,
D: Tiempo de ejecución;
- (9): $Y = (1- C/D)$,
C: tiempo de vinculo de OS,
D: Tiempo de ejecución

Atributo al que ayuda a calcular: Eficiencia de almacenamiento

Granularidad: Modulo, sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica a disciplinas de diseño e implementación, primero nivel de modulo y después a nivel de sistema

MÉTRICA 400

Nombre en español: Lista de chequeo de control de acceso

Nombre en inglés: Access control checklist

Descripción: La métrica es la suma de puntajes de los siguientes elementos aplicables, dividido por el número de elementos aplicables:

1. Controles de acceso de E/S de usuario provistos: Los requerimientos para el control de acceso de usuarios debe ser identificados durante la fase de requerimientos. Provisiones para comprobación de identificación y contraseña deben ser diseñados e implementados para cumplir con los requerimientos. Esta es una medida binaria aplicada a todas las tres fases e identifica la atención prestada en esa área
2. Control de acceso de la base de datos provisto: esta es una medida binaria que identifica si los requerimientos para los controles de la base de datos han sido especificados, diseñados y las capacidades implementadas. ejemplos de controles de acceso a la base de datos son tablas de autorización y cerraduras de privacidad,
3. Protección de memoria a través de tareas: similar a los puntos (1) y (2), esta medida identifica la progresión desde las declaraciones de requerimientos a la implementación de protección de memoria a través de tareas. ejemplos de este tipo de protección, comúnmente suministran algún grado por el sistema operativo, son tareas preventivas a partir de invocación de otras tareas, tareas de acceso de sistemas de registro y el uso de comandos privilegiados.

Fórmula: A nivel de sistema:

$$X=A/B,$$

A: total de puntajes desde los elementos aplicables

B: Número de elementos aplicables;

Atributo al que ayuda a calcular: Control de acceso

Granularidad: sistema

Referencias: McCall [18]-1

Espectro: Gris

Otros: Aplica a disciplinas de análisis, diseño, implementación a nivel de sistemas

MÉTRICA 401

Nombre en español: Atributos locales

Nombre en inglés: Local attributes

Descripción: es una cuenta de los atributos definidos localmente en una clase (esos componentes inmediatamente definidos en el registro de etiqueta). Esta

medida sin embargo es similar en naturaleza a la MÉTRICA de "number of attributes", es importante porque caracteriza la diferencia (delta) entre el número total de atributos definidos y el número local de atributos definidos. El tamaño de delta puede indicar una clase que es "compleja" (un alto índice de de atributos locales del total de atributos), o estrechamente acoplada a sus antecesores (un bajo índice de de atributos locales del total de atributos).

Fórmula:

Atributo al que ayuda a calcular: Tamaño

Granularidad: clase

Referencias: Pritchett [69]

Espectro: Gris

Otros: similar a la métrica de "number of attributes"(métrica 44)

MÉTRICA 402

Nombre en español: Operaciones totales

Nombre en inglés: Total operations = Number of operations

Descripción: es una cuenta del número de operaciones primitivas definidas para una clase. Esto incluye las operaciones heredadas pero no anuladas y esas operaciones localmente definidas. Esta medida es importante porque incrementa la Funcionalidad igual al número de operaciones incrementadas, así incrementando la probabilidad de fallas.

Fórmula:

NO=El número total de operaciones definidas por el usuario dentro del sistema, incluyendo métodos y funciones globales

Atributo al que ayuda a calcular: Tamaño

Granularidad: Clase

Referencias: Pritchett [69], Lanza [88]

Espectro: Gris

Otros: Puede ser igual a "NOM- Number of methods" (métrica 273)

MÉTRICA 403

Nombre en español: Operaciones locales

Nombre en inglés: Local operations

Descripción: es una cuenta del número total de operaciones primitivas definidas localmente para la clase (definida en el mismo paquete en el cual el tipo de etiqueta está definido). Sin embargo es similar en naturaleza a la métrica "total operations", el número de operaciones locales es importante porque este caracteriza la diferencia (delta) entre el número de operaciones totales definidas y las operaciones locales definidas. el tamaño de delta puede indicar una clase que es "compleja" (un alto índice de de operaciones locales del total de operaciones), o estrechamente acoplada a sus antecesores (un bajo índice de de operaciones locales del total de operaciones).

Fórmula:

LO=El número total de operaciones locales definidas en la clase

Atributo al que ayuda a calcular: Tamaño

Influencia sobre otros atributos:

Granularidad: Clase

Referencias: Pritchett [69]

Espectro: Gris

Otros: Puede tener relación con las métricas 273 y 402

MÉTRICA 404

Nombre en español: Operaciones remplazadas

Nombre en inglés: Overridden Operations

Descripción: Es una cuenta del número de operaciones definidas en un antecesor, pero invalidada por la clase. Las operaciones invalidadas pueden complicar la Comprensibilidad de una clase y pueden también proveer una oportunidad para errores.

Fórmula:

OO= número de operaciones o métodos que han sido remplazados en la clase

Atributo al que ayuda a calcular: Comprensibilidad

Granularidad: Clase

Referencias: Pritchett [69]

Espectro: Gris

Otros: Puede tener relación con la métrica "Number of methods overridden" (métrica 132)

MÉTRICA 405

Nombre en español: Operaciones de clase amplia

Nombre en inglés: Class-Wide Operations

Atributo al que ayuda a calcular: Cohesión

Granularidad: Clase

Referencias: Pritchett [69]

Espectro: Oscuro

MÉTRICA 406

Nombre en español: Atributos de tipos abstractos de datos

Nombre en inglés: Attributes of Abstract Data Types -ADTs

Atributo al que ayuda a calcular: Cohesión

Referencias: PRITCHETT [69], Li [72]

Espectro: Oscuro

MÉTRICA 407

Nombre en español: Tipos abstractos de datos referenciados

Nombre en inglés: Abstract Data Types Referenced

Atributo al que ayuda a calcular: Cohesión

Referencias: PRITCHETT [69]

Espectro: oscuro

MÉTRICA 408

Nombre en español: Promedio de parámetros por método

Nombre en inglés: Average parameters per Method - APM

Descripción: Definida como el número total de parámetros en la clase / número total de elementos. Lorenz y Kidd sugieren que no debe exceder a 0.7

Fórmula:

$$\text{APM} = \text{total de parámetros en la clase} / \text{número total de elementos}$$

Atributo al que ayuda a calcular: Tamaño

Granularidad: Clase

Referencias: El-Wakil [71]

Espectro: Gris

MÉTRICA 409

Nombre en español: Número de punto y comas (NOSC)

Nombre en inglés: NOSC

Descripción: esta métrica intenta definir el tamaño de la clase similar a LOC. La definición no indica si sólo los puntos y comas correspondientes a las declaraciones de los métodos que deben contarse o punto y coma correspondientes a las declaraciones también deben ser incluidas. Este indicador es aplicable muy tarde en el ciclo de desarrollo y no es útil para la predicción de esfuerzo.

Fórmula:

$$\text{NOSC} = \text{Número de punto y coma}$$

Atributo al que ayuda a calcular: Tamaño

Granularidad: Clase

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica a la disciplina de implementación. Puede ser la misma de "Size 1" (métrica 284)

MÉTRICA 410

Nombre en español: Número de atributos y métodos (NAANM)

Nombre en inglés: NAANM

Descripción: Este indicador captura el número de propiedades de la clase. Estos a su vez están determinados por el comportamiento intrínseco de la clase. No está claro si los métodos locales deberían incluir tanto métodos público y los no-públicos de la clase o no. Suponemos que lo hacen. Este indicador puede ser calculado antes, cuando el diseño detallado de la clase ha terminado. Uno debe tener en cuenta que los valores determinados en la fase de diseño detallado y después de la implementación de la clase tienden a cambiar a medida que las variables locales, utilizados para la ejecución, puede recibir contados como miembros de datos y el número de métodos puede cambiar. En los sistemas orientados a objetos, las clases son definidas y aplican de forma incremental mediante el uso de la Herencia. Desde el punto de vista de este esfuerzo de definición parece correcta, pero no refleja el

tamaño de tiempo de ejecución de los objetos de una instancia de una definición de clase. Este indicador no es compatible con la Reutilización dentro de la clase. Métodos, desarrollados para excluir código común entre otros métodos, tienden a aumentar el valor de la métrica

Fórmula:

NAANM= Número de atributos + Número de métodos locales

Atributo al que ayuda a calcular: Tamaño

Influencia sobre otros atributos: Complejidad

Granularidad: Clase

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica a disciplinas de diseño, implementación, puede ser la misma de "Size 2" (métrica 285)

MÉTRICA 411

Nombre en español: Número ponderado de atributos y métodos (WNAAWNMM)

Nombre en inglés: WNAAWNMM

Descripción: Este indicador es similar al sistema métrica "NAANM" (métrica 410). Es una medida más fina de la Complejidad interna, porque los métodos y los atributos de las personas son ponderados de acuerdo con una noción subjetiva de la Complejidad. El uso de pesas para determinar una mejor medida de un comportamiento de la clase hace que sea difícil volver a definir la aplicación de métricas en las primeras etapas del ciclo de desarrollo.

Atributo al que ayuda a calcular: Complejidad

Granularidad: Clase

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica a disciplinas diseño, implementación.

MÉTRICA 412

Nombre en español: Complejidad de atributo

Nombre en inglés: Attribute complexity -AC

Descripción: La definición de los usos de pesos asignados estáticamente. En este indicador requiere información acerca de los atributos definidos en una clase y por lo tanto sólo puede ser utilizado al final del ciclo de desarrollo.

Fórmula:

$AC = \sum$ de todos los atributos de la clase R (i), donde

R (i)= Complejidad del atributo i de la clase definida estáticamente

Atributo al que ayuda a calcular: Complejidad

Granularidad: Clase

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica a disciplina diseño

MÉTRICA 413

Nombre en español: Complejidad operacional

Nombre en inglés: Operational Complexity - OC

Descripción: La semántica de una clase es establecida por las operaciones que pueden realizarse en el objeto o por el objeto. Esto caracteriza el comportamiento de cada clase. El comportamiento de la clase es a su vez determinado por sus responsabilidades. Es difícil hacer una medición aproximada de la métrica sobre la base de las responsabilidades de clase, identificados antes en el ciclo de desarrollo, porque requiere una forma de asignar valores a la Complejidad de cada responsabilidad. En la fase de análisis de los primeros modelos son aún exploratorios y por lo tanto es demasiado temprano para cualquier asignación.

Fórmula:

$$OC = \sum \text{de todas las operaciones de la clase } O(i), \text{ donde}$$
$$O(i) = \text{valor de la Complejidad de la operación } i \text{ escogida de la clase definida estáticamente}$$

Atributo al que ayuda a calcular: Complejidad

Granularidad: Clase

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica a disciplina diseño

MÉTRICA 414

Nombre en español: Complejidad de argumento de operación

Nombre en inglés: Operation argument complexity - OAC

Descripción: Este indicador es calculado sobre la base de operación. El funcionamiento exacto de los métodos de interfaz está determinada en detalle en la fase de diseño. En este indicador también requiere de la asignación de valores de la Complejidad de los argumentos, y por lo tanto no puede ser re-implimentado en base la información disponible en la fase de análisis.

Fórmula:

$$OAC = \sum \text{de todos los argumentos para las operaciones de una clase } P(i),$$

donde

$$P(i) = \text{Complejidad de las operaciones del argumento } i \text{ escogida de una pre-}$$

definida

Atributo al que ayuda a calcular: Complejidad

Granularidad: Clase

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica a disciplina diseño

MÉTRICA 415

Nombre en español: Métrica de Cohesión

Nombre en inglés: Cohesion metric - CM

Descripción: Esta incluyen tanto el argumento y los tipos de cambio de parámetros. El número de conjuntos disjuntos mostrar la heterogeneidad de

las operaciones. Este indicador puede aplicarse cuando la firma de todos los métodos es clara. Por lo tanto, sólo puede aplicarse desde la fase de diseño detallado. Similitud tipos de parámetros "es una afirmación débil para la medición de la Cohesión. En el caso de sobrecarga de funciones de dos firmas del mismo método puede tener el tipo discontinuo de parámetros de los métodos de una clase

Fórmula:

$$CM=M/N * 100, \text{ donde}$$

M: Número de conjuntos disjuntos formado por la intersección de la serie N de la argumentación y tipos de retorno de los métodos de una clase,

N: Número de conjuntos de argumentos y tipos de retorno de todos los métodos de una clase

Atributo al que ayuda a calcular: Cohesion

Granularidad: clase

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica a disciplina diseño

MÉTRICA 416

Nombre en español: Número de métodos locales privados

Nombre en inglés: Number of private local methods - NPM

Descripción: Este indicador sólo puede calcularse en la fase de diseño de clase. También pasa por alto los logros obtenidos a través de la Reutilización de factorización de código común entre los métodos. En las primeras etapas no tenemos una descripción interna de las clases y por lo tanto este indicador no puede ser utilizado en las primeras etapas

Fórmula:

$$NPM= \text{número de métodos no públicos de una clase}$$

Atributo al que ayuda a calcular: Complejidad

Granularidad: Clase

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica a disciplina diseño

MÉTRICA 417

Nombre en español: Ponderado de clases abstractas a cociente de atributo (WACAAR)

Nombre en inglés: Weighted Abstract Class to Attribute Ratio - WACAAR

Descripción: Esta métrica capta la noción de que una clase puede estar compuesta de muchas nociones conceptuales separadas. Las clases abstractas son generalmente definidas como la raíz de un árbol de jerarquía de clases. La métrica no define claramente lo que entiende por clases abstractas. Este indicador puede ser aplicado en la primera fase de diseño de clase y no sirve para volver a la aplicación en la fase de análisis

Fórmula:

$$WACAAR = A/B, \text{ donde}$$

A: Número de clases abstractas definidas en la clase,

B: Número total de atributos

Atributo al que ayuda a calcular: Complejidad

Granularidad: Clase

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica a disciplina diseño

MÉTRICA 418

Nombre en español: Ponderado de métodos privados a cociente ponderado de métodos (WPMMR)

Nombre en inglés: Weighted Private Methods to Weighted Methods Ratio - WPMMR

Descripción: La métrica requiere que todos los métodos para asignar pesos basados en una idea predeterminada de la Complejidad. Como la información sobre los métodos privados no está disponible hasta la etapa de diseño de clase este indicador tampoco puede aproximarse en la fase de análisis

Fórmula:

$$WPMMR = \{\sum A\} / \{\sum B\}, \text{ Donde}$$

A: Pesos de los métodos privados,

B: pesos de todos los métodos

Atributo al que ayuda a calcular: Complejidad

Granularidad: Clase

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica a disciplina diseño

MÉTRICA 419

Nombre en español: Número de estado estables en una clase (NSSC)

Nombre en inglés: Number of Stable States of a Class - NSSC

Descripción: La definición de clase tiene lugar en la fase de análisis y de diseño. Sin embargo, la información de estado de una clase puede o no puede ser documentado en la fase de análisis. Por lo tanto, este indicador es una metodología específica y puede aplicarse en las fases de análisis, si la información acerca de la naturaleza con estado de la clase está disponible

Fórmula:

$$NSSC = \text{Número de estados estables de una clase}$$

Atributo al que ayuda a calcular: Complejidad

Granularidad: Clase

Referencias: OJHA [78]

Espectro: Clase

Otros: Aplica a disciplinas de análisis y diseño

MÉTRICA 420

Nombre en español: Número de eventos distintos en una clase (NDEC)

Nombre en inglés: Number of Distinct Events in a Class - NDEC

Descripción: Eventos causa del estado de transición. Estado de información de transición para una clase dada puede o no estar disponible en la etapa de análisis. Por lo tanto, este indicador es una metodología específica y puede aplicarse en la fase de análisis, si la información acerca de la naturaleza con estado de la clase está disponible.

Fórmula:

$$\text{NDEC} = \text{Número de eventos distintos en la clase}$$

Atributo al que ayuda a calcular: Complejidad

Influencia sobre otros atributos: tamaño

Granularidad: Clase

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica a disciplinas de análisis y diseño

MÉTRICA 421

Nombre en español: Tamaño del dominio del mensaje

Nombre en inglés: Message Domain Size - MDS

Descripción: Este indicador trata de medir el número de distintos tipos de mensajes a los que el objeto puede responder. Esto depende del comportamiento de la clase esperada por sus usuarios. La definición de métricas requiere información acerca de la naturaleza con estado de la clase. Esto depende de la metodología utilizada para construir los sistemas orientados a objetos

Fórmula:

$$\text{MDS} = \text{Número de métodos que afectan el estado de un objeto}$$

Atributo al que ayuda a calcular: tamaño

Influencia sobre otros atributos: Complejidad

Granularidad: Clase

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica a disciplina de diseño

MÉTRICA 422

Nombre en español: Número de vagos

Nombre en inglés: Number of Tramps - NT

Descripción: Los parámetros ajenos son los que no tengan acceso a los métodos de clase. Estos tienden a dar una impresión errónea acerca de la transformación efectuada por el método y hacer que sea difícil para los clientes a comprender una clase. La información sobre los parámetros que son utilizados y cuáles no, sólo puede determinarse una vez que la aplicación ha terminado. Por lo tanto, este indicador no puede aplicarse en las primeras etapas del ciclo de desarrollo.

Fórmula:

NT= número de parámetros ajenos en las firmas de los métodos de una clase

Atributo al que ayuda a calcular: Complejidad

Influencia sobre otros atributos:

Granularidad: Clase

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica a disciplina de implementación

MÉTRICA 423

Nombre en español: Porción de los distintos eventos del Número de métodos de interface

Nombre en inglés: Ratio of Distinct Event to Number of Interface Methods (REIM)

Descripción: Este indicador Aplica generalmente en la fase de diseño final. Pero, si la información de estado acerca de una clase está disponible en la fase de análisis, entonces esto podría volver a ser implementado utilizando las responsabilidades como una aproximación a los métodos de interfaz

Fórmula:

REIM= proporción de NDEC para el número de métodos de la interfaz

Atributo al que ayuda a calcular: Complejidad

Granularidad: Clase

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica a disciplina de diseño. Relación con métrica 420

MÉTRICA 424

Nombre en español: Tamaño del vocabulario del mensaje

Nombre en inglés: Message Vocabulary Size - MVS

Descripción: Esta medida indica el número de mensajes distintos que un programador debe conocer. Conjunto de todos los mensajes enviados por una clase sólo puede determinarse en la etapa de ejecución. En el comienzo de los modelos de sistema, este indicador puede ser aproximado por el número total de las responsabilidades de todas las clases con las que colabora de una clase

Fórmula:

MVS= Número de mensajes diferentes enviados por una clase

Atributo al que ayuda a calcular: Tamaño

Granularidad: Clase

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica a disciplina de implementación

MÉTRICA 425

Nombre en español: Métrica de acoplamiento de operación

Nombre en inglés: Operation Coupling metrics - OCMi

Descripción: Estas medidas sólo pueden aplicarse en la implementación de la clase etapa en la que todas las operaciones de las operaciones de las interacciones son claras. Las operaciones incluyen tanto métodos públicos como no-pública. El punto importante en estos indicadores es que las interacciones están relacionadas con una dirección. La dirección está determinada por que el quien accede a quien. Estas direcciones son utilizados para calificar la interacción. En el cliente-servidor de direcciones perspectiva de determinar quién es el cliente y que es el servidor. las clases de co-operación son los clientes de unos a otros y por lo tanto debe ser considerado como más bien acoplados

Fórmula:

OCM1= número de operaciones de una clase que accede a otras clases.

OCM2= número de operaciones de una clase que son accedidas por otras clases.

OCM3= número de operaciones de una clase que co-operan con otras clases

Atributo al que ayuda a calcular: Acoplamiento

Granularidad: Clase

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica a disciplina de implementación

MÉTRICA 426

Nombre en español: Métrica de acoplamiento de clase

Nombre en inglés: Class Coupling metrics - CCMi

Descripción: Los patrones de acceso de la clase pueden determinar por la relación de asociación determinada en el análisis y la etapa de diseño de alto nivel. Por lo tanto, esta es una métrica candidato para la re-aplicación en la fase de análisis.

Fórmula:

CCM1= número de clases accedidas por una clase.

CCM2= número de clases accesando una clase.

CCM3= número de clases co-operando con una clase

Atributo al que ayuda a calcular: Acoplamiento

Granularidad: Clase

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica a disciplinas de análisis y diseño

MÉTRICA 427

Nombre en español: Violaciones a la ley de Demeter

Nombre en inglés: Violations of Law of Demeter - VOD

Descripción: Esta medidas esta basada n en los principios de la Ley de Deméter o del buen estilo (en programación orientada a objetos). Al igual que

otras medidas similares número exacto de los mensajes enviados sólo puede determinarse en el momento de la implementación de clase

Fórmula:

VOD= Número de mensajes enviados a clases que no son parámetros para mensajes o explícitamente declarados dentro de la clase

Atributo al que ayuda a calcular:

Influencia sobre otros atributos:

Granularidad: Clase

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica a disciplina de implementación

MÉTRICA 428

Nombre en español: Número total de clases

Nombre en inglés: Total Number of Classes - TNC

Descripción: Esta métrica mide el número total de los conceptos constitutivos de un sistema. Un sistema con muchas clases es más difícil de componer que uno con menor número de clases. Experimentos con el esfuerzo de desarrollo no han sido realizados con una contabilidad separada para la construcción de componentes y los esfuerzos de la composición. Por lo tanto, no puede determinarse si el mayor esfuerzo requerido para la construcción de sistemas con mayor número de clases también es debido al mayor esfuerzo de composición.

Fórmula:

TNC= Número total de clases en el sistema

Atributo al que ayuda a calcular: Tamaño

Granularidad: Sistema

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica en disciplina de análisis. Similar a métrica 274

MÉTRICA 429

Nombre en español: Número de clases de máquina de estado finito

Nombre en inglés: Number of Finite State Machine Classes - NFSM

Descripción: Las clases que instigan acción son clasificadas como máquinas de estado finito (FSM). Más número de esas clases implica que habrá mayor cantidad de mensajes entre los componentes. Esto hace que sea difícil de integrar el sistema y por lo tanto debe aumentar el esfuerzo de composición

Fórmula:

NFSM= número de clases que envían mensajes fuera

Atributo al que ayuda a calcular: Tamaño

Influencia sobre otros atributos:

Granularidad: Sistema

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica en disciplina de análisis

MÉTRICA 430

Nombre en español: Número de clases de tipo abstracto de datos

Nombre en inglés: Number of Abstract Data Types Classes - NADT

Descripción: Las clases que no inician ninguna acción son clasificadas como de Tipos de datos abstractos. Ellos sólo sufren las acciones. Estas clases son más fáciles de implementar e integrar con el sistema.

Fórmula:

NADT= Número de clases que solo reciben mensajes

Atributo al que ayuda a calcular: Tamaño

Granularidad: Sistema

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica en disciplina de análisis

MÉTRICA 431

Nombre en español: Número de máquina de estado finito – interacciones maquina de estado finito

Nombre en inglés: Number of FSM - FSM Interactions -NFSMI

Descripción: Este indicador esta basada en el supuesto de que la interacción entre dos clases designadas como máquinas de estado finito es más difícil de entender que las interacciones entre una clase designada como Finite State Machine y un tipo abstracto de datos

Fórmula:

NFSMI = El número total de pares de colaborador formado disjuntos, donde ambas clases pueden ser clasificados como FSM

Atributo al que ayuda a calcular: Tamaño

Granularidad: Sistema

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica en disciplina de análisis

MÉTRICA 432

Nombre en español: Número de máquina de estado finito – interacciones tipo abstracto de dato

Nombre en inglés: Number of FSM-ADT Interactions - NADTI

Descripción: complemento de la MÉTRICA NFSMI

Fórmula:

NADTI = El número total de pares de colaborador formado disjuntos, donde ambas clases pueden ser clasificados como ADT

Atributo al que ayuda a calcular: Tamaño

Granularidad: Sistema

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica en disciplina de análisis

MÉTRICA 433

Nombre en español: Número de clases agrupadas

Nombre en inglés: Number of Class Clusters - NCC

Descripción: La métrica NCC mide el número de interconexiones entre los sistemas, así como el número de grupos de los conceptos en el sistema.

Fórmula:

NCC= Número de conjuntos disjuntos formados por la intersección de clases asociadas con cada clase

Atributo al que ayuda a calcular: tamaño

Granularidad: Sistema

Referencias: OJHA [78]

Espectro: Oscuro

Otros: Aplica en disciplina de análisis

MÉTRICA 434

Nombre en español: Asociación de Complejidad

Nombre en inglés: Association Complexity - AC

Descripción: es similar a la métrica de McCabe [92] (Cyclomatic complexity)

Fórmula:

AC= Número de aristas en el gráfico de colaboración - número de nodos en el gráfico de colaboración + (2 * Número de piezas desconectado)

Atributo al que ayuda a calcular: Complejidad

Granularidad: Sistema

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica en disciplina de análisis

MÉTRICA 435

Nombre en español: Número total de contratos

Nombre en inglés: Total Number of Contracts - TC

Descripción: Esta métrica mide el número total de posibles escenarios de servidor de cliente en el sistema

Fórmula:

TC=número de posibles escenarios

Atributo al que ayuda a calcular: Complejidad

Granularidad: Sistema

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica en disciplina de análisis

MÉTRICA 436

Nombre en español: Cuenta de responsabilidad para una clase

Nombre en inglés: Responsibility Count for a Class - RCC

Descripción: Este indicador es similar al tamaño métrico de mensajes de dominio (MDS). Si es definido a nivel suficiente de detalle cada uno, la responsabilidad debería traducirse en al menos un método distinto. Por lo tanto, el RCC refleja el número de mensajes distintos a los que una clase puede responder.

Fórmula:

RCC=Número de responsabilidades de una clase listadas en la tarjeta CRC

Atributo al que ayuda a calcular: tamaño

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica en disciplina de análisis (Tarjetas CRC)

MÉTRICA 437

Nombre en español: Responsabilidades heredadas por una clase

Nombre en inglés: Inherited Responsibility Set for a Class - IRSC

Descripción: En un modelo de Herencia de la clase cumple estrictamente todas las responsabilidades de sus clases padre. Paradigma orientado a objetos de las clases de permisos para volver a poner en práctica las conductas heredadas. Sea F la función que determina el número de responsabilidades heredadas que son re-implementado en una clase. Como 'F' no está claro en la etapa de análisis, sólo contar el número total de las responsabilidades que la clase hereda. El 'las responsabilidades heredadas' son calculadas atravesando la clase de estructuras jerárquicas en reversa y sin contar con cualquier clase de responsabilidad más de una vez.

Fórmula:

IRSC= Suma de los valores RCC de todas las clases de la clase hereda dada.

Atributo al que ayuda a calcular: Tamaño

Granularidad:

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica en disciplina de análisis (Tarjetas CRC), Esta relacionada con la métrica 436

MÉTRICA 438

Nombre en español: Cuenta de responsabilidades normalizadas por una clase

Nombre en inglés: métrica combina las responsabilidades de una clase que son definidos a nivel local y una que hereda. Esta es la opinión de la clase que es visto por sus clientes. Intuitivamente, NRCC determina el número total de comportamiento que un cliente de una clase puede solicitar. Los primeros modelos son más abstractos y menos detallados. Por lo tanto, cada uno la responsabilidad define un comportamiento distinto. Este indicador da una medida peor de los casos del número de comportamientos de clase distinta de la que el usuario de una clase debe ser consciente

Fórmula:

NRCC= RCC+IRSC

Atributo al que ayuda a calcular: Tamaño

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica en disciplina de análisis (Tarjetas CRC). Esta relacionada con las métricas 436 y 2437

MÉTRICA 439

Nombre en español: Número de tipos colaboradores

Nombre en inglés: Number of Collaborator Types - NCT

Descripción: Esta métrica mide la cantidad de diferentes abstracciones conceptuales con las que el desarrollador de clase necesita interactuar. Dado que cada jerarquía de Herencia representa la especialización de una determinada construcción, el promotor tiene mucho más trabajo que hacer si dos colaboradores proceden de las jerarquías separadas de si procede de la misma jerarquía.

Fórmula:

$$NCT = \text{Número de estructuras jerárquicas disjuntas pertenecientes a los colaboradores}$$

Atributo al que ayuda a calcular: Complejidad

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica en disciplina de análisis (Tarjetas CRC)

MÉTRICA 440

Nombre en español: Número de colaboradores distintos

Nombre en inglés: Number of Distinct Collaborators - NDC

Descripción: Este indicador contabiliza el número total de colaboradores que una clase tiene. Como cada una de ellas representa una conducta definida, esta métrica mide el número de diferentes tipos de comportamientos de un desarrollador de clase debe estar al corriente

Fórmula:

$$NDC = \text{Número de clases distintas que colaboran con la clase dada}$$

Atributo al que ayuda a calcular: Complejidad

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica en disciplina de análisis (Tarjetas CRC)

MÉTRICA 441

Nombre en español: Promedio de tamaño de vocabulario colaborador por responsabilidad

Nombre en inglés: Average Collaborator Vocabulary Size per Responsibility - ACVSR

Descripción: Esta métrica mide la fuerza de la dependencia conductista. Tenga en cuenta que las clases producidas más de una vez como

colaboradores, en la tarjeta de CRC, son contadas varias veces. Cabe señalar que las tarjetas CRC no explícitamente todas las diferentes relaciones entre las clases. Las relaciones entre las clases pueden ser la especialización, la agregación y de asociación. Debido a que el enfoque responsabilidades en la descripción externa de las clases, supone que la mayoría de los colaboradores tienen una relación de asociación con la clase.

Fórmula:

$$ACVSR_{clase} = \frac{\{\sum \text{colaboradores generales de (NRCCcolaborador)}\}}{RCC_{clase}}$$

Atributo al que ayuda a calcular: Complejidad

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica en disciplina de análisis (Tarjetas CRC), relación con las métricas 438 y 436

MÉTRICA 442

Nombre en español: Métrica de Cohesión de clase

Nombre en inglés: Class Cohesion metric -CCM

Descripción: Esta métrica favorece la encapsulación. El comportamiento de las dependencias introduce el acoplamiento entre las clases y disminuye la encapsulación. Esto aumenta la dificultad de implementar una clase. Kowele ha observado que los sistemas orientados a objetos tienden a tener más de acoplamiento, debido a que componentes reutilizables pueden ser desarrollados por la factorización de las responsabilidades comunes. Por lo tanto, es aconsejable que las métricas de dependencia y la cohesión debe aplicarse junto con las métricas de Reutilización.

Fórmula:

$$CCM = \frac{\text{Porcentaje de responsabilidades cumplidas a cabalidad por la clase}}{\text{misma}}$$

Atributo al que ayuda a calcular: Complejidad

Referencias: OJHA [78]

Espectro: Gris

Otros: Tiene relación con "LCOM"- (métrica 21-A)

MÉTRICA 443

Nombre en español: Número de clases cliente

Nombre en inglés: Number of client classes - NCC

Descripción: Este indicador puede utilizarse para establecer si los beneficios de la Reutilización, aunque a costa de aumentar el acoplamiento, son suficientes para justificar la clase. Clases sin los clientes no pueden ser definidas en el sistema, pero están presentes en el diseño para ayudar a construir las jerarquías de clase.

Fórmula:

$$NCC = \text{Número de clases distintas que usan los servicios de una clase dada}$$

Atributo al que ayuda a calcular: Tamaño

Granularidad: Sistema
Referencias: OJHA [78]
Espectro: gris
Otros: Aplica en disciplina de análisis (Tarjetas CRC)

MÉTRICA 444

Nombre en español: Promedio de cuenta de responsabilidades de clase (ARCC)

Nombre en inglés: Average Responsibility Count of Classes - ARCC

Descripción: Estos dos indicadores ponen en evidencia la distribución de los tamaños de las clases que componen el sistema. Esta distribución, junto con los promedios da al administrador una mejor vista del sistema. Por ejemplo, un diseño en que una clase tiene 90% de los métodos no sería muy orientado a objetos y que probablemente sería mucho más difícil de codificar y depurar que una de los métodos distribuidos más uniformemente". Estas cifras están basadas en este razonamiento.

Fórmula:

$$ARCC = \{\sum \text{todas las clases de (RCC)}\} / TNC$$

Atributo al que ayuda a calcular: tamaño

Granularidad: Sistema

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica en disciplina de análisis (Tarjetas CRC) relación co las métrica 436 y 428

MÉTRICA 445

Nombre en español: Desviación estándar de cuenta de responsabilidades de clases (SDRCC)

Nombre en inglés: Standard Deviation of Responsibility Count of Classes - SDRCC

Descripción: Estos dos indicadores ponen en evidencia la distribución de los tamaños de las clases que componen el sistema. Esta distribución, junto con los promedios da al administrador una mejor vista del sistema. Por ejemplo, un diseño en que una clase tiene 90% de los métodos no sería muy orientado a objetos y que probablemente sería mucho más difícil de codificar y depurar que una de los métodos distribuidos más uniformemente". Estas cifras están basadas en este razonamiento.

Fórmula:

$$SDRCC = \text{SQRT} \{ [\sum \text{todas las clases de (RCC-ARCC)}^2] / TNC \}$$

Atributo al que ayuda a calcular: Tamaño

Granularidad: Sistema

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica en disciplina de análisis (Tarjetas CRC). Relación con las métricas 436, 428 y 444

MÉTRICA 446

Nombre en español: Colaboraciones totales (TC)

Nombre en inglés: Total Collaborations - TC

Descripción: Estas métricas miden los diferentes aspectos de las interacciones entre las clases. La interacción entre las clases es producida debido a las dependencias de comportamiento entre ellos. Cuanto más las dependencias entre las clases, más serán las interacciones entre ellos. Esto es capturado por la métrica de TC. Cabe señalar que un valor muy alto de este indicador puede indicar un sistema más abstracto. Las métricas ANDC y SDNDC miden la distribución de las interacciones entre las clases

Fórmula:

$$TC = \sum \text{sobre todas las clases de (NDC)}$$

Atributo al que ayuda a calcular: Complejidad

Granularidad: Sistema

Referencias: OJHA [78]

Espectro: Gris,

Otros: Aplica en disciplina de análisis (Tarjetas CRC). Relación con la métrica 440

MÉTRICA 447

Nombre en español: Número promedio de distintas colaboraciones de clases (ANDC)

Nombre en inglés: Average Number of Distinct Collaborators of Classes - ANDC

Descripción: Estas métricas miden los diferentes aspectos de las interacciones entre las clases. La interacción entre las clases es producida debido a las dependencias de comportamiento entre ellos. Cuanto más las dependencias entre las clases, más serán las interacciones entre ellos. Esto es capturado por la métrica de TC. Cabe señalar que un valor muy alto de este indicador puede indicar un sistema más abstracto. las métricas ANDC y SDNDC miden la distribución de las interacciones entre las clases

Fórmula:

$$ANDC = \{ \sum \text{sobre todas las clases de (NDC)} \} / TNC$$

Atributo al que ayuda a calcular: Complejidad

Granularidad: Sistema

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica en disciplina de análisis (Tarjetas CRC), relación con las métrica 440 y 428

MÉTRICA 448

Nombre en español: Desviación estándar de Número promedio de distintas colaboraciones de clases (SDNDC)

Nombre en inglés: Standard Deviation of Number of Distinct Collaborators of Classes - SDNDC

Descripción: Estas métricas miden los diferentes aspectos de las interacciones entre las clases. La interacción entre las clases es producida debido a las dependencias de comportamiento entre ellos. Cuanto más las dependencias entre las clases, más serán las interacciones entre ellos. Esto es capturado por la métrica de TC. Cabe señalar que un valor muy alto de este indicador puede indicar un sistema más abstracto. las métricas ANDC y SDNDC miden la distribución de las interacciones entre las clases

Fórmula:

$$SDNDC = \text{SQRT} \{ [\sum \text{todas las clases de (NDC-ANDC)}^2] / \text{TNC} \}$$

Atributo al que ayuda a calcular: Complejidad

Granularidad: Sistema

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica en disciplina de análisis (Tarjetas CRC) relación con las métrica 440, 447 y 428

MÉTRICA 449

Nombre en español: Número de clases del tipo de máquina de estado finito (NFSM)

Nombre en inglés: Number of classes of type Finite State Machine (NFSM)

Descripción: Las clases que instigan acción son clasificadas como máquinas de estado finito (FSM). Más número de esas clases implica que habrá mayor cantidad de mensajes entre los componentes. Esto hace que sea difícil de integrar el sistema y por lo tanto debe aumentar el esfuerzo de composición

Fórmula:

$$NFSM = \text{número de clases que envían mensajes fuera}$$

Atributo al que ayuda a calcular: composición

Granularidad: Sistema

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica en disciplina de análisis (Tarjetas CRC)

MÉTRICA 450

Nombre en español: Número de clases del tipo: tipo abstracto de datos (NADT)

Nombre en inglés: Number of classes of type Abstract Data Type (NADT)

Descripción: Las clases que no inician ninguna acción son clasifican como Tipos de datos abstractos. Ellos sólo sufren las acciones. Estas clases son más fáciles de implementar e integrar con el sistema.

Fórmula:

$$NADT = \text{Número de clases que solo reciben mensajes}$$

Atributo al que ayuda a calcular: tamaño

Influencia sobre otros atributos: Abstracto

Granularidad: Sistema

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica en disciplina de análisis (Tarjetas CRC)

MÉTRICA 451

Nombre en español: Número de clases jerárquicas (NCH)

Nombre en inglés: Number of Class Hierarchies - NCH

Descripción: Esta métrica capta el número de tipos de concepto distintos de construcción del sistema. El uso de esta métrica favorece la construcción de las jerarquías de clase. Todas las clases que no pertenecen a ninguna estructura jerárquica establecida son consideradas que son una parte de la jerarquía de clases integrado por sí misma.

Fórmula:

NCH = Número de estructuras jerárquicas de clase a la que todas las clases que componen el sistema pueden ser asignadas.

Atributo al que ayuda a calcular: jerarquía

Granularidad: Sistema

Referencias: OJHA [78]

Espectro: Gris

Otros: Aplica en disciplina de análisis (Tarjetas CRC)

MÉTRICA 452

Nombre en español: Instalabilidad

Nombre en inglés: Instability - I

Descripción: La inestabilidad entre los paquetes (I) mide la relación entre la salida y el número total de entrada- y los acoplamientos de salida de clases dentro del paquete de clases fuera del paquete (de acoplamiento a las clases de clases externas del paquete, que utiliza la definición de acoplamiento). Cada clase cuenta sólo una vez. Cero, si el paquete no contiene las clases o si las clases externas no son utilizadas por las clases del paquete. I es aplicable principalmente a sistemas orientados a objetos

Fórmula:

$$I = Ce / (Ca + Ce),$$

Ca: Acoplamiento aferente, el cual es el número de clases fuera del paquete que dependen de clases dentro del paquete (Eg. referencias de entrada).

Ce: acoplamiento eferente, el cual es el número de clases fuera del paquete, de las cuales las clases dentro del paquete dependen

Atributo al que ayuda a calcular: Abstracción

Granularidad: paquete

Rango: Rango (0,1), I=0 indica un paquete con Estabilidad máxima. I=1, indica un paquete con inEstabilidad máxima.

Referencias: Andersson[74], Martin [30]

Espectro: Gris

Otros: relacionado con la métrica 25

MÉTRICA 453

Nombre en español: Acoplamiento de Abstracción de datos de paquetes

Nombre en inglés: Package Data Abstraction Coupling - PDAC

Descripción: Sube la métrica DAC a nivel de paquete. Representa el número de ADTse referenciadas de una clase a otro cruza fronteras del paquete. Cuanto mayor sea el número, el más estricto es el acoplamiento de dos paquetes. Cuentan son los campos definidos en clases dentro de un paquete básico (paquete de nivel superior), haciendo referencia a un tipo definido por el usuario en otro paquete de base. Paquetes básicos que, lógicamente, pertenecen al mismo componente son considerados como un solo paquete. Los valores son números enteros que van desde 0 (que indica que no existe referencia a ningún otro ADT) a un número máximo en una escala absoluta. Este indicador es calculado en una gráfica de paquete que contiene las declaraciones de paquete, la clase y los nodos de campo (nodos AST), así como contiene y las relaciones de tipo de referencia (bordes).

Atributo al que ayuda a calcular: acoplamiento

Referencias: Panas [96]

Espectro: Gris

Otros: Relación con la métrica 28

MÉTRICA 454

Nombre en español: Cambio de dependencia entre las clases

Nombre en inglés: Change Dependency between Classes - CDBC

Descripción: determina la cantidad potencial de trabajo de seguimiento que existe en una clase de cliente cuando es modificada una clase de servidor. Indica la fuerza de acoplamiento. El objetivo es mantener los cambios en el sistema local para la reducción del acoplamiento del sistema. Los valores más bajos indican un menor acoplamiento y, por tanto, una mejor Estabilidad en el sistema. El valor CDBC es definido entre las clases cliente y un servidor como el número de métodos que necesitan ser (potencialmente) cambiados en el cliente si cambia de servidor. El valor de CDBC está entre 0 y el conteo de métodos de la clase. Calculamos el valor de CDBC promedio de cada clase (cliente) sobre todas las clases (servidor) que está directamente conectada. La escala es racional. Este indicador es calculado en un programa de información gráfica conteniendo clase, método, constructor, campo, y los nodos de inicialización del bloque (nodos AST) y (además de la contención sintáctica) el tipo de referencia, extensiones y relaciones implementadas (los bordes).

Fórmula:

$$CDBC (CC, SC) = \min (n, A).$$

$$A = \sum \text{de número de accesos a la implementación } i \text{ de } (\alpha_i) + (1+k) * \sum \text{de accesos a la interface } i \text{ de } (\alpha_i).$$

SC: clase del servidor que de cambia

CC: clase dependiente del cliente,

α : número de métodos de CC potencialmente afectados por un cambio,

k: corresponde a la Estabilidad de la interface de SC
Atributo al que ayuda a calcular: acoplamiento
Granularidad: clase
Referencias: Panas [96], Martin [30], Bar [75]
Espectro: Gris

MÉTRICA 455

Nombre en español: Ajuste de cohesión en la clase
Nombre en inglés: Tight Class Cohesion - TCC
Descripción: es el número relativo de los métodos directamente conectados en una clase. TCC indica el grado de conectividad entre los métodos visibles en una clase. Los valores más altos indican una mayor cohesión de las clases. Los valores bajos indican que una clase tiene una baja cohesión. Este indicador es calculado sobre el gráfico de llamadas (calculado por defecto).

Fórmula:

$$TCC = NDC / NP.$$

NP = $[n \times (n-1)] / 2$, NP es el máximo número de conexiones directas o indirectas en una clase n: métodos locales en una clase.

NDC el número conexiones directas en la clase.

El TCC para una clase es 0 si NP = 0.

Atributo al que ayuda a calcular: Cohesion
Granularidad: clase
Rango: El rango de valores resultantes 0,0 a 1,0
Referencias: Panas [96], Bieman [97], Bar [75]
Espectro: Gris
Otros: Comparada con LCOM (métrica 21-A)

MÉTRICA 456

Nombre en español: Ajuste de cohesión en el paquete
Nombre en inglés: Tight Package Cohesion - TPC
Descripción: levanta la métrica TCP a nivel de paquete y recupera el número relativo de las clases conectados directamente por las llamadas en un paquete. TPC indica el grado de conectividad entre las clases dentro de un paquete. La métrica TPC es calculada similar a la PDAC en los paquetes de base que contenga los componentes. Los valores más altos indican una mayor cohesión del conjunto. Este indicador es calculado en una gráfica llamada con las declaraciones del paquete.

Fórmula:

Teniendo en cuenta el número n del paquete de clases locales,

TPC es definido como NDP sobre NP, donde

NP = $[n \times (n-1)] / 2$ los pares posibles de estas clases y el

NDP: el número de parejas de clases realmente llamando a otra.

El TPC de un paquete es 0 si NP = 0.

Atributo al que ayuda a calcular: Cohesion
Granularidad: Paquete

Rango: El rango de valores resultantes 0,0 a 1,0 en una escala racional.

Referencias: Panas [96]

Espectro: Gris

MÉTRICA 457

Nombre en español: Perdida de cohesión en la clase

Nombre en inglés: Loose class cohesion

Descripción: es el número relativo de métodos conectados directa o indirectamente

Fórmula:

$$LCC = (NDC + NIC) / NP.$$

NP = $[n \times (n-1)] / 2$, NP es el máximo número de conexiones directas o indirectas en una clase n: métodos locales en una clase.

NDC: el número conexiones directas en la clase.

NIC: el número conexiones indirectas en la clase.

Atributo al que ayuda a calcular: Cohesion

Granularidad: clase

Referencias: Bieman [97]

Espectro: Gris

MÉTRICA 458

Nombre en español: Falta de documentación

Nombre en inglés: Lack of Documentation - LOD

Descripción: mide la cantidad de declaraciones de indocumentadas por clase (cuenta las declaraciones de la clase en sí y las declaraciones de método, pero no las declaraciones de campo). Sólo el estilo de documentación Javadoc es tenido en cuenta. Documentación dentro de los métodos es ignorada. Sólo la sintaxis de los comentarios es analizada, no la semántica. El valor límite de detección es calculado para cada clase o interfaz como el número de declaraciones de indocumentados. Un LOD de 0 indica que están documentadas todas las entidades posibles, un valor más alto indica que falta documentación. El límite de detección es en una escala absoluta. Un problema con este indicador es aparentemente generación Javadoc automática. Herramientas de apoyo a esta función impiden la medición exacta de las declaraciones de indocumentadas. Sin embargo, este problema implica simplemente que no podemos descubrir todo el código indocumentado, es decir, no podemos recuperar todos los problemas inherentes.

Fórmula:

$$LOD = \text{cantidad de declaraciones sin documentar}$$

Atributo al que ayuda a calcular: tamaño

Influencia sobre otros atributos: Documentación

Referencias: Panas [96]

Espectro: Gris

MÉTRICA 459

Nombre en español: Localidad de los datos

Nombre en inglés: Locality of data - LD

Descripción: Representa un atributo directamente conectado con la calidad de la Abstracción contenida por una clase

Fórmula:

$$LD = \frac{\{\sum \text{desde } l=1 \text{ hasta } n \text{ de } (L_i)\}}{\{\sum \text{desde } i=1 \text{ hasta } n \text{ de } (T_i)\}},$$

Donde

Mi: Métodos de la clase (Excluyendo todos los métodos escritura/lectura para variables instancia).

Li (1<=i<=n): conjunto de variables "locales" accesados por Mi (directamente o vía métodos de escritura/lectura). Estos son: variables de instancia no pública de la clase C, variables de instancia protegida heredada de su superclase, variables estáticas definidas localmente en Mi. Ti (1<=i<=n): conjunto de variables usadas en Mi, excepto por variables locales no estáticas definidas en Mi.

Atributo al que ayuda a calcular: acoplamiento

Granularidad: Clase

Referencias: Martin [30], Bar [75]

Espectro: Gris

MÉTRICA 460

Nombre en español: Densidad de defectos

Nombre en inglés: Defect density

Fórmula:

$$DD = \text{Número de defectos conocidos/tamaño del producto (LOC)}$$

Atributo al que ayuda a calcular: tamaño

Referencias: Fenton [77]

Espectro: Gris

MÉTRICA 461

Nombre en español: Promedio de parámetros del método

Nombre en inglés: average method parameters - APPM

Fórmula:

$$APPM = \text{Número de métodos de parámetro/ número total de métodos}$$

Atributo al que ayuda a calcular: Tamaño

Referencias: Genero [90]

Espectro: Gris

MÉTRICA 462

Nombre en español: Acoplamiento para componentes súper especializados

Nombre en inglés: coupling to overspecialised components

Descripción: La Reutilización reduce si una función, caso de prueba, o altstep es ejecutada en un componente que es ejecutado en un componente principal también, pero está ligado a otro más especializado. Por lo tanto, esta métrica esta relacionada con el número de tales casos a que el número total de funciones, casos de prueba, y altsteps acoplados a los componentes en general.

Fórmula:

$$X=A/B,$$

A: número de entidades de comportamiento innecesariamente ejecutadas en un componente especializado,

B: número total de entidades de comportamiento ejecutadas en los componentes

Atributo al que ayuda a calcular: Acoplamiento

Referencias: Zeiss [56]

Espectro: Gris

MÉTRICA 463

Nombre en español: Tasa de cobertura

Nombre en inglés: Coverage ratio

Descripción: El porcentaje de la Funcionalidad deseada que está presente en el producto de software.

Atributo al que ayuda a calcular: Idoneidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 464

Nombre en español: Porcentaje de cobertura reducido

Nombre en inglés: Scaled coverage percentage

Descripción: El porcentaje de la Funcionalidad preferido que está realmente presente en el producto de software, indexada según la importancia relativa de la Funcionalidad requerida para la ejecución de la actividad empresarial. Nota: Los protocolos de medición muestran que esta característica está relacionada con la especificación funcional. Esta característica puede ser de importancia cuando una herramienta es seleccionada de un conjunto de productos de software estándar.

Atributo al que ayuda a calcular: Idoneidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 465

Nombre en español: Cociente de la especificación funcional de cambio

Nombre en inglés: Functional specification change ratio

Descripción: La relación de las funciones que han sido cambiadas (el cambio incluye la adición, modificación y supresión).

Atributo al que ayuda a calcular: Idoneidad

Referencias: QUINT2 [58] [59]
Espectro: Oscuro

MÉTRICA 466

Nombre en español: Relación de solicitud de mejoramiento
Nombre en inglés: Improvement request ratio
Descripción: El número de peticiones de mejora de las funciones de software de los usuarios por mes después del parto
Atributo al que ayuda a calcular: Idoneidad
Referencias: QUINT2 [58] [59]
Espectro: Oscuro

MÉTRICA 467

Nombre en español: Relación de reclamación
Nombre en inglés: Claim ratio
Descripción: El número de reclamaciones de los usuarios por mes y por kilo de línea del código fuente
Atributo al que ayuda a calcular: Idoneidad
Referencias: QUINT2 [58] [59]
Espectro: Oscuro

MÉTRICA 468

Nombre en español: Cociente de fracaso
Nombre en inglés: Failure ratio
Descripción: La proporción de transacciones procesadas incorrecta al total de las operaciones presentadas
Atributo al que ayuda a calcular: Exactitud
Referencias: QUINT2 [58] [59]
Espectro: Oscuro

MÉTRICA 469

Nombre en español: Cociente de dígitos significativos
Nombre en inglés: Significant digits ratio
Descripción: La relación de las cifras significativas aplicadas a los dígitos necesarios significativas
Atributo al que ayuda a calcular: Exactitud
Referencias: QUINT2 [58] [59]
Espectro: Oscuro

MÉTRICA 470

Nombre en español: Relación de Conformidad con el manual
Nombre en inglés: Manual conformance ratio
Descripción: La relación de las funciones desarrolladas y los productos que correspondan a las funciones escritas en los manuales del usuario

Atributo al que ayuda a calcular: Exactitud

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 471

Nombre en español: Cociente del tratamiento de redondeo

Nombre en inglés: Rounding treatment ratio

Descripción: La relación de funciones con el tratamiento necesario para redondear el número total de las funciones implementadas

Atributo al que ayuda a calcular: Exactitud

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 472

Nombre en español: Esfuerzo por la interacción

Nombre en inglés: Effort per interaction

Descripción: El esfuerzo necesario para alcanzar la Interoperabilidad por unidad de tamaño de la Interoperabilidad

Atributo al que ayuda a calcular: Interoperabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 473

Nombre en español: Cociente de formato de datos coincidente

Nombre en inglés: Matched data format ratio

Descripción: La relación de los formatos de datos adaptados a los del otro sistema en la interacción

Atributo al que ayuda a calcular: Interoperabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 474

Nombre en español: Cociente de caracteres coincidentes

Nombre en inglés: Matched character ratio

Descripción: La proporción de caracteres gráficos y caracteres de control equiparable a los de los otros sistemas de interacción

Atributo al que ayuda a calcular: Interoperabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 475

Nombre en español: Cociente de interfaces coincidentes

Nombre en inglés: Matched interface ratio

Descripción: La relación de las interfaces de coincidir con las de los otros sistemas de interacción

Atributo al que ayuda a calcular: Interoperabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 476

Nombre en español: Proporción de estándar observado

Nombre en inglés: Observed standard ratio

Descripción: La relación de las normas observadas a las normas introducidas en el sistema o entre el sistema

Atributo al que ayuda a calcular: Interoperabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 477

Nombre en español: Proporción de formato de datos estandarizado

Nombre en inglés: Standardised data format ratio

Descripción: La relación de los formatos de datos normalizados a los formatos de datos para ser normalizados

Atributo al que ayuda a calcular: Conformidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 478

Nombre en español: Proporción de formato de medio estandarizado

Nombre en inglés: Standardised medium format ratio

Descripción: La relación de los formatos de medio normalizado para los formatos de medio a ser normalizados

Atributo al que ayuda a calcular: Conformidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 479

Nombre en español: Proporción de carácter estandarizado

Nombre en inglés: Standardised character ratio

Descripción: La proporción de los caracteres gráficos normalizados y caracteres de control a las que se habrán normalizado

Atributo al que ayuda a calcular: Conformidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 480

Nombre en español: Proporción de interface estandarizada

Nombre en inglés: Standardised interface ratio

Descripción: La relación de las interfaces estandarizadas para las interfaces de ser normalizados

Atributo al que ayuda a calcular: Conformidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 481

Nombre en español: Relación de expresión de Conformidad

Nombre en inglés: Expression conformance ratio

Descripción: La relación de las funciones que tienen expresiones específicas o métodos de cálculo adaptado a las reglas o normas.

Atributo al que ayuda a calcular: Conformidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 482

Nombre en español: Resistencia

Nombre en inglés: Resistance

Descripción: Estimación de la probabilidad de que una cierta cantidad de esfuerzo (tiempo, dinero, equipo) las mediciones de software de seguridad no será dejado de lado

Atributo al que ayuda a calcular: Seguridad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 483

Nombre en español: Resistencia a Hacker

Nombre en inglés: Hacker-resistance

Descripción: Un grupo de hackers reconocido no podrá atentar contra el producto de software dentro de una nueva definición de período de tiempo, mientras que el acceso físico al producto de software no existe. Nota: Estos indicadores sólo se ocupan de las mediciones de software. La seguridad puede mejorarse considerablemente las mediciones físicas y de organización.

Atributo al que ayuda a calcular: Seguridad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 484

Nombre en español: Relación de cifrado de datos

Nombre en inglés: Ciphered data ratio

Descripción: La relación de los datos cifrados a los datos a ser cifrado

Atributo al que ayuda a calcular: Seguridad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 486

Nombre en español: Relación de historia de acceso

Nombre en inglés: Access history ratio

Descripción: La relación de la información confidencial que tienen historias de acceso a toda la información confidencial

Atributo al que ayuda a calcular: Seguridad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 486

Nombre en español: Relación de daño de datos

Nombre en inglés: Data damage ratio

Descripción: ¿Cuántas veces por mes fue de los datos dañados?

Atributo al que ayuda a calcular: Seguridad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 487

Nombre en español: Esfuerzo de control de operación

Nombre en inglés: Operation control effort

Descripción: La cantidad de tiempo pierdido durante el procesamiento debido a las actividades de control de funcionamiento, manual o automáticamente

Atributo al que ayuda a calcular: Trazabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 488

Nombre en español: Facilidad de control de operación

Nombre en inglés: Ease of operation control

Descripción: La cantidad de esfuerzo necesario para realizar el control de la operación

Atributo al que ayuda a calcular: Trazabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 489

Nombre en español: Densidad de fallas del producto

Nombre en inglés: Product fault density

Descripción: La relación de número de defectos en un producto liberado a la unidad de volumen (por ejemplo, páginas, Kloc) de un producto liberado (por ejemplo, los documentos del usuario, el código fuente)

Atributo al que ayuda a calcular: Madurez

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 490

Nombre en español: Estabilidad del producto

Nombre en inglés: Product stability

Descripción: La proporción de faltas corregido hasta un punto de tiempo para la estimación del número de fallas se presentan inicialmente en un producto

Atributo al que ayuda a calcular: Madurez

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 491

Nombre en español: Densidad de prueba

Nombre en inglés: Test density

Descripción: La proporción del volumen de ensayo (por ejemplo, número de casos de prueba) llevó a cabo durante la fase de desarrollo a la unidad de volumen de un producto probado y puesto en libertad.

Atributo al que ayuda a calcular: Madurez

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 492

Nombre en español: Disturbios

Nombre en inglés: Disturbances

Descripción: El número de tiempos de proceso fue detenido debido al uso incorrecto en un plazo determinado.

Atributo al que ayuda a calcular: Tolerancia a fallos

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 493

Nombre en español: Vulnerabilidad

Nombre en inglés: Vulnerability

Descripción: La medida en que es posible detener el procesamiento debido a un uso incorrecto

Atributo al que ayuda a calcular: Tolerancia a fallos

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 494

Nombre en español: Valor de integridad

Nombre en inglés: Integrity value

Descripción: Estimación de la probabilidad de una infracción que se especifica en un plazo determinado

Fórmula:

$$I = \sum \text{desde } i=1 \text{ hasta } n \text{ de } [(1 - \text{threat}(i)) * (1 - \text{security}(i))].$$

Donde

I: es un tipo de uso infríngete o incorrecto,

threat (i) es la probabilidad de uso incorrecto de tipo i que ocurrirá con un cierto periodo,

security (i) es la Efectividad de las decisiones de diseño de seguridad que puede prevenir uso incorrecto del tipo i, expresado como el valor entre 0 y 1

Atributo al que ayuda a calcular: Tolerancia a fallos

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 495

Nombre en español: Proporción de detección de error en la operación/entrada

Nombre en inglés: Operational/input error detection ratio

Descripción: La relación del número de operaciones erróneas o insumos detectados por un sistema de software para el número de operaciones erróneas o insumos a cabo durante un período determinado de tiempo

Atributo al que ayuda a calcular: Tolerancia a fallos

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 496

Nombre en español: Tiempo medio de reparación

Nombre en inglés: Mean time to repair (MTTR)

Descripción: El tiempo medio necesario para recuperar el producto de software dañado

Atributo al que ayuda a calcular: Recuperabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 497

Nombre en español: Tiempo medio de interrupción

Nombre en inglés: Mean breakdown time

Descripción: El tiempo medio entre la aparición de fallo del software del sistema y la finalización de la recuperación y reinicio de la degradación.

Atributo al que ayuda a calcular: Recuperabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 498

Nombre en español: Mean Tiempo medio de reinicio

Nombre en inglés: Mean restart time

Descripción: El tiempo medio entre la conclusión de recuperación de la ruptura del software del sistema y reinicie después de completar avería

Atributo al que ayuda a calcular: Recuperabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 499

Nombre en español: Vida media de la falla

Nombre en inglés: Mean fault life

Descripción: La media de tiempo entre la introducción de una falta y la eliminación de la falla

Atributo al que ayuda a calcular: Recuperabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

Otros:

MÉTRICA 500

Nombre en español: Completa relación de la disponibilidad de tiempo

Nombre en inglés: Full-time availability ratio

Descripción: La proporción de tiempo en el que el producto de software no esta en recuperación durante un período determinado, que es el MTBF / (MTBF + MTTR). Nota: Este indicador sólo es válida para los sistemas en total disponibilidad de tiempo es deseable. Lo contrario, la recuperación puede tener lugar cuando el sistema no es necesario que esté disponible.

Atributo al que ayuda a calcular: Disponibilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 501

Nombre en español: Relación de disponibilidad relativa

Nombre en inglés: Relative availability ratio

Descripción: La proporción de tiempo que el producto de software está disponible para el momento en que es necesitado. Nota: Este indicador difiere de la proporción de tiempo completo, cuando la disponibilidad de la disponibilidad de tiempo completo no es necesaria y la recuperación puede tener lugar cuando el sistema no es necesario que esté disponible.

Atributo al que ayuda a calcular: Disponibilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 502

Nombre en español: Tiempo de retroceso

Nombre en inglés: Backsliding time

Descripción: Por pre-definidos desglose, el tiempo necesario para restablecer el funcionamiento del producto degradado. Nota: Cuando es necesario el tiempo de retroceso a de ser extremadamente corto, un producto de software degradado (por ejemplo, una versión anterior) puede mantenerse en espera.

Atributo al que ayuda a calcular: Degradabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 503

Nombre en español: Tasa de Comprensibilidad

Nombre en inglés: Rated understandability

Descripción: La Comprensibilidad de las instrucciones, los menús, los comandos, pictogramas, iconos, la información de ayuda, instrucciones, manuales, etc., del producto de software según la puntuación otorgada por el usuario.

Atributo al que ayuda a calcular: Comprensibilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 504

Nombre en español: Puntuación de legibilidad

Nombre en inglés: Readability score

Descripción: Evaluación de la legibilidad de los productos de software (los mensajes en pantalla, documentos, pictogramas, etc.) Nota: Este indicador no tiene nada que ver con el significado de un texto, sino que simplemente indica el uso de palabras y la longitud de la frase.

Atributo al que ayuda a calcular: Comprensibilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 505

Nombre en español: Concepto de claridad

Nombre en inglés: Concept clearness

Descripción: La proporción de las funciones que pueden explicarse mediante el uso de claro, los modelos familiares para ilustrar los conceptos. Esto representa el grado en que las funciones y las convenciones de un producto de software se explican a través de modelos utilizando conceptos familiares en el mundo cotidiano.

Atributo al que ayuda a calcular: Comprensibilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 506

Nombre en español: Disponibilidad de demostración de software

Nombre en inglés: Availability of demonstration software

Descripción: La proporción de las funciones presentaron a los usuarios a través de software de demostración.

Atributo al que ayuda a calcular: Comprensibilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 507

Nombre en español: Claridad de uso

Nombre en inglés: Usage clearness

Descripción: La relación de las funciones explicadas o mediante el uso de modelos claros o presentadas al usuario a través de software de demostración o de todos modos descritos.

Atributo al que ayuda a calcular: Comprensibilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 508

Nombre en español: Disponibilidad de datos de entrada / salida de elementos de la lista

Nombre en inglés: Availability of input/output data items list

Descripción: Para las funciones con las operaciones de entrada o de salida, el número de entrada / salida de los artículos que figuran en esta lista.

Atributo al que ayuda a calcular: Comprensibilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 509

Nombre en español: Habilidad de reconocer parámetros modificables

Nombre en inglés: Recognisability of modifiable parameters

Descripción: La proporción de los parámetros que son identificados como sea modificado o fijo.

Atributo al que ayuda a calcular: Comprensibilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 510

Nombre en español: Tiempo promedio de aprendizaje

Nombre en inglés: Average learning time

Descripción: El tiempo promedio de un fin de usuario desde el grupo objetivo tiene que aprender a trabajar con el producto de software, además de la cantidad de tiempo que necesita orientación.

Atributo al que ayuda a calcular: Facilidad de aprendizaje

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 511

Nombre en español: Usabilidad del manual

Nombre en inglés: Usability of manual

Descripción: La proporción de ejemplos, las entradas de índice, ilustraciones y tablas por comando, y / o la proporción de referencias, capítulos, secciones y subtítulos por página. Nota: Es útil para prever si el usuario puede usar el manual. Un rango de número de ejemplos, las entradas de índice, etc., por la línea de comandos puede ser especificado.

Atributo al que ayuda a calcular: Facilidad de aprendizaje

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 512

Nombre en español: Disponibilidad del manual

Nombre en inglés: Availability of manual

Descripción: El grado de disponibilidad de los manuales de referencia, manuales de usuario y los documentos de auto-enseñanza, tales como manuales de operación, materiales de referencia de gramática, manual de instalación, etc., para las funciones de software.

Atributo al que ayuda a calcular: Facilidad de aprendizaje

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 513

Nombre en español: Disponibilidad de las funciones de pre-aprendizaje

Nombre en inglés: Availability of pre-learning functions

Descripción: Para un determinado rango de funciones de software, el grado de disponibilidad de materiales para la pre-aprendizaje (antes de la utilización del sistema), tales como manuales de auto-enseñanza, tutoriales automatizados o instrucciones.

Atributo al que ayuda a calcular: Facilidad de aprendizaje

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 514

Nombre en español: Disponibilidad de las funciones de aprendizaje

Nombre en inglés: Availability of pre-learning functions

Descripción: Para un determinado rango de funciones de software, el grado de disponibilidad de materiales para la pre-aprendizaje (antes de la utilización del sistema), tales como manuales de auto-enseñanza, tutoriales automatizados o instrucciones.

Atributo al que ayuda a calcular: Facilidad de aprendizaje

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 515

Nombre en español: Disponibilidad de las funciones de ayuda

Nombre en inglés: Availability of help functions

Descripción: Para un determinado rango de funciones de software, la relación entre el número de funciones de ayuda disponibles y el número de funciones de ayuda necesaria.

Atributo al que ayuda a calcular: Facilidad de aprendizaje

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 516

Nombre en español: Tasa de errores de usuario

Nombre en inglés: Rate of user errors

Descripción: La proporción de reclamos de los usuarios como resultado de operaciones incorrectas causadas por falta de comprensión de los programas o manuales.

Atributo al que ayuda a calcular: Facilidad de aprendizaje

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 517

Nombre en español: Tiempo requerido para aprender una operación

Nombre en inglés: Time required to learn operation

Descripción: El tiempo de uso desde el primer uso hasta el uso en el que el tiempo de operación se acorta a X / Y de la primera utilización para realizar la tarea específica mismo.

Atributo al que ayuda a calcular: Facilidad de aprendizaje

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 518

Nombre en español: Esfuerzo requerido para aprender una operación

Nombre en inglés: Effort required to learn one operation

Descripción: La proporción de tiempo necesario para aprender una operación para una tarea específica y el tiempo de operación.

Atributo al que ayuda a calcular: Facilidad de aprendizaje

Referencias: QUINT2 [58] [59]
Espectro: Oscuro

MÉTRICA 519

Nombre en español: Juicio de experto sobre operatividad
Nombre en inglés: Expert judgement on operability
Descripción: La medida en que el producto de software presenta una Funcionalidad para el usuario, sin obstáculos, a juzgar por un equipo de expertos en este campo.
Atributo al que ayuda a calcular: Operatividad
Referencias: QUINT2 [58] [59]
Espectro: Oscuro

MÉTRICA 520

Nombre en español: Operatividad comparada con la muestra
Nombre en inglés: Operability compared with sample
Descripción: La operatividad del producto de software es comparado con la operatividad de un producto de muestra predeterminado.
Atributo al que ayuda a calcular: Operatividad
Referencias: QUINT2 [58] [59]
Espectro: Oscuro

MÉTRICA 521

Nombre en español: Operatividad en la practica
Nombre en inglés: Operability in practice
Descripción: La medida en que el producto de software presenta una Funcionalidad para el usuario, sin obstáculos, a juzgar por los usuarios después de un período de uso.
Atributo al que ayuda a calcular: Operatividad
Referencias: QUINT2 [58] [59]
Espectro: Oscuro

MÉTRICA 522

Nombre en español: Configuración del tiempo de operación
Nombre en inglés: Set-up installation time
Descripción: La cantidad de tiempo requerido para el proceso de instalación, incluidos los preparativos, la ejecución de conjunto con el medio ambiente y la verificación.
Atributo al que ayuda a calcular: Operatividad
Referencias: QUINT2 [58] [59]
Espectro: Oscuro

MÉTRICA 523

Nombre en español: Puesta en marcha procedimientos de instalación con la interacción humana

Nombre en inglés: Set-up installation procedures with human interaction

Descripción: El número de pasos para la puesta en marcha de instalación de operación que requiere la interacción humana (para el uso diario).

Atributo al que ayuda a calcular: Operatividad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 524

Nombre en español: Disponibilidad de puesta en marcha de rendimiento

Nombre en inglés: Availability of set-up performance

Descripción: La relación del número de set-up actuaciones disponibles y el número total de actuaciones: la disponibilidad de puesta en marcha de instalación, reinicie, la disponibilidad de funciones de configuración de confirmación de la instalación.

Atributo al que ayuda a calcular: Operatividad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 525

Nombre en español: Facilidad de configurar

Nombre en inglés: Ease of set-up

Descripción: El tiempo medio requerido para crear el software o sistema.

Atributo al que ayuda a calcular: Operatividad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 526

Nombre en español: Disponibilidad de puesta en marcha de instalación reinicie

Nombre en inglés: Availability of set-up installation restart

Descripción: El número de puntos en los que los usuarios pueden hacer una pausa y de la que la operación de instalación puede reiniciarse.

Atributo al que ayuda a calcular: Operatividad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 527

Nombre en español: Disponibilidad de puesta en marcha de instalación de preparación

Nombre en inglés: Availability of set-up installation preparation

Descripción: El número de tipos y la cantidad de recursos informáticos, incluido el sistema, hardware, software y personal a ser empleados para la instalación.

Atributo al que ayuda a calcular: Operatividad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 528

Nombre en español: Disponibilidad de puesta en marcha de instalación de confirmación

Nombre en inglés: Availability of set-up installation confirmation

Descripción: El número de pasos para la operación de instalación de confirmación de que son necesarios para validar que el software o el sistema ha sido instalado correctamente y está listo para ser utilizado de manera adecuada.

Atributo al que ayuda a calcular: Operatividad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 529

Nombre en español: Proporción del valor por defecto la disponibilidad

Nombre en inglés: Default value availability ratio

Descripción: La relación de los comandos que operan con valores por defecto al número total de órdenes de explotación.

Atributo al que ayuda a calcular: Operatividad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 530

Nombre en español: Uniformidad de comandos

Nombre en inglés: Command uniformity

Descripción: La proporción de los comandos de funcionamiento que los modelos uniformes, que esta basada n en el sentido común y las reglas comprensibles.

Atributo al que ayuda a calcular: Operatividad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

Otros:

MÉTRICA 531

Nombre en español: Consistencia en términos del mensaje

Nombre en inglés: Consistency of terms in message

Descripción: La proporción de términos de mensajes del sistema que están estandarizados.

Atributo al que ayuda a calcular: Operatividad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 532

Nombre en español: Claridad de mensaje

Nombre en inglés: Message clearness

Descripción: La proporción de mensajes del sistema de software o sistema en el que las causas y las medidas correspondientes están claramente identificados por el usuario que recibe los mensajes.

Atributo al que ayuda a calcular: Operatividad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 533

Nombre en español: Adaptabilidad al nivel de dificultad

Nombre en inglés: Skill level adaptability

Descripción: La proporción de las funciones para las que los métodos de funcionamiento pueden seleccionarse para la correspondiente al nivel del usuario de la habilidad.

Atributo al que ayuda a calcular: Operatividad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 534

Nombre en español: Uniformidad de la pantalla de manipulación de operaciones

Nombre en inglés: Uniformity of screen manipulating operations

Descripción: La proporción de tipos de pantalla, usando la manipulación de las operaciones de los convenios básicos comunes o patrones.

Atributo al que ayuda a calcular: Operatividad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 535

Nombre en español: Estabilidad de áreas de entrada/salida en pantalla

Nombre en inglés: Stability of input/output areas on display

Descripción: La proporción de entrada / salida de formatos de pantalla diseñados con formatos normalizados en el que la posición y la forma de entrada / salida de los campos son comúnmente establecidos.

Atributo al que ayuda a calcular: Operatividad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 536

Nombre en español: Número de pulsaciones del teclado

Nombre en inglés: Number of keystrokes

Descripción: El número de pulsaciones de teclado de operación requerido por el usuario para llevar a cabo el trabajo. Esos son los golpes de teclas, haga clic en clave, haga clic en el botón, la pantalla táctil, mover la pluma, mover el ratón, etc.

Atributo al que ayuda a calcular: Operatividad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 537

Nombre en español: Disponibilidad de esfuerzo reducido para operaciones repetidas

Nombre en inglés: Availability of reduced effort for repeated operations

Descripción: La proporción de golpes necesarios para repetir la operación de los trazos necesarios para la primera operación a realizar una tarea específica.

Atributo al que ayuda a calcular: Operatividad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 538

Nombre en español: Tiempo medio entre operaciones de error humano

Nombre en inglés: Mean time between human error operations

Descripción: Intervalo de tiempo promedio entre un error de operación (operación de errores humanos) y la siguiente.

Atributo al que ayuda a calcular: Operatividad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 539

Nombre en español: Hora de apagado de la operación

Nombre en inglés: Ultimate operation time

Descripción: El tiempo requerido para la operación, que se reduce en última instancia y no puede ser reducido por más mejoras.

Atributo al que ayuda a calcular: Operatividad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 540

Nombre en español: Proporción de disponibilidad de la función guía

Nombre en inglés: Guide function availability ratio

Descripción: La relación de las funciones de guía a disposición de los requeridos para un determinado conjunto de funciones.

Atributo al que ayuda a calcular: Operatividad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 541

Nombre en español: Proporción de habilidad de cancelación de la operación de error humano

Nombre en inglés: Human error operation cancellability ratio

Descripción: La relación de mando / entradas de datos que puede ser cancelado con el comando total / entradas de datos.

Atributo al que ayuda a calcular: Operatividad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 542

Nombre en español: Habilidad para enfatizar expresiones

Nombre en inglés: Ability to emphasise expressions

Descripción: La relación del número de puesto efectivamente en práctica los medios necesarios a los que se proporcionan a insistir en expresiones para el usuario para un determinado conjunto de funciones. Se considera que el color, el brillo de sonido, y la animación son medios para insistir en las expresiones.

Atributo al que ayuda a calcular: Operatividad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 543

Nombre en español: Tiempo de respuesta para el usuario

Nombre en inglés: Response time for user

Descripción: El tiempo transcurrido desde la solicitud de entrada del usuario o un comando más cercano a la respuesta a esa solicitud por el software o sistema. La primera respuesta a la petición puede ser un estado o informe expresando que el software o el sistema están procesando la petición. Al menos tiempo promedio, mínimo y máximo deben ser medidos.

Atributo al que ayuda a calcular: Operatividad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 544

Nombre en español: Tiempo de pantalla

Nombre en inglés: Display time

Descripción: El tiempo transcurrido desde la pantalla actual en la pantalla después de solicitar el cambio, a la siguiente pantalla completa en la pantalla. Al menos tiempo promedio, mínimo y máximo deben ser medidos.

Atributo al que ayuda a calcular: Operatividad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 545

Nombre en español: Juicio de expertos sobre la explicitud

Nombre en inglés: Expert judgement on explicitness

Descripción: Explicitación de los productos de software, a juzgar por un equipo de expertos.

Atributo al que ayuda a calcular: Explicitud

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 546

Nombre en español: Tiempo inseguro

Nombre en inglés: Insecure time

Descripción: El período de tiempo que el usuario está inseguro acerca de la situación del producto de software.

Atributo al que ayuda a calcular: Explicitud

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 547

Nombre en español: Proporción del reporte de estado

Nombre en inglés: Status report ratio

Descripción: La relación de todas las acciones posibles, que es seguido inmediatamente por un informe de situación del producto de software después de la activación.

Atributo al que ayuda a calcular: Explicitud

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 548

Nombre en español: Tiempo inseguro en la practica

Nombre en inglés: Insecure time in practice

Descripción: El período de tiempo que el usuario está inseguro acerca de la situación del producto de software, reportados por los usuarios después de un período de uso.

Atributo al que ayuda a calcular: Explicitud

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 549

Nombre en español: Reporte de progreso o estado de la proporción de disponibilidad

Nombre en inglés: Status or progress report availability ratio

Descripción: La relación de estado o los informes de progreso que están a disposición de los requeridos para un determinado conjunto de funciones.

Atributo al que ayuda a calcular: Explicitud

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 550

Nombre en español: Proporción de configurabilidad

Nombre en inglés: Configurability ratio

Descripción: La relación de las partes de la Funcionalidad que puede ser cambiado sin cambiar el código con el número total de partes de la Funcionalidad.

Atributo al que ayuda a calcular: Personalización

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 551

Nombre en español: Esfuerzo de de configurabilidad

Nombre en inglés: Configurability effort

Descripción: El esfuerzo necesario para cambiar la Funcionalidad del producto de software. Nota: Esto sólo se puede medir con el producto de software instalado.

Atributo al que ayuda a calcular: Personalización

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 552

Nombre en español: Juicio de usuario sobre el atractivo

Nombre en inglés: User judgement on attractivity

Descripción: El nivel de equipamiento de los productos de software, especialmente la disponibilidad de servicios adicionales, el comportamiento y la presentación, a juzgar por el usuario.

Atributo al que ayuda a calcular: Personalización

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 553

Nombre en español: Proporción de la función de reconocimiento

Nombre en inglés: Function recognition ratio

Descripción: Relación de las funciones que un usuario inexperto promedio distingue en el producto de software sin ayuda, dentro de un período de tiempo.

Atributo al que ayuda a calcular: Claridad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 554

Nombre en español: Proporción de la función de usuario

Nombre en inglés: Function use ratio

Descripción: Relación de las funciones del producto de software que un usuario medio en realidad emplea después de un período de tiempo determinado.

Atributo al que ayuda a calcular: Claridad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 555

Nombre en español: Proporción de texto expuesto

Nombre en inglés: Ratio of expounding text

Descripción: La proporción de la cantidad de exponer el texto (incluyendo mensajes de error) disponibles en el producto de software a la cantidad total de textos que pueden ser presentados en la pantalla.

Atributo al que ayuda a calcular: Ayuda

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 556

Nombre en español: Proporción normalizada de texto expuesto

Nombre en inglés: Normalised ratio of expounding text

Descripción: La proporción de la cantidad de exponer el texto (incluyendo mensajes de error) disponibles en el producto de software para el tamaño del producto de software. Nota: Los protocolos de medición sólo cuenta la cantidad de exponer el texto, no para mejorar la legibilidad o Comprensibilidad de la misma. Por lo tanto, estos indicadores son sólo mínimamente válidos. En lugar de texto, los protocolos de medición también pueden ser utilizados para los iconos, sonidos, etc. Cuando se usa con ese fin, los acuerdos tienen que hacerse sobre la forma de medir e interpretar estos indicadores.

Atributo al que ayuda a calcular: Ayuda

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 557

Nombre en español: Tasa de amigabilidad

Nombre en inglés: Rated user-friendliness

Descripción: Proporción de usuarios que prefieren el producto de software nuevo en los últimos, después de un cierto período de uso práctico.

Atributo al que ayuda a calcular: Amigabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 558

Nombre en español: Juicio de experto sobre amigabilidad

Nombre en inglés: Expert judgement on user-friendliness

Descripción: La Facilidad de uso, a juzgar por un equipo de expertos en temas como: la composición de la pantalla, el vocabulario, la aplicación del color y el sonido.

Atributo al que ayuda a calcular: Amigabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 559

Nombre en español: Amigabilidad comparada con la muestra

Nombre en inglés: User-friendliness compared to sample

Descripción: Facilidad de uso en comparación con un producto de muestra, a juzgar por un equipo de expertos. Los expertos de decidir en qué medida los usuarios de uso del producto de software coinciden con el producto de la muestra.

Atributo al que ayuda a calcular: Amigabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 560

Nombre en español: Tiempo de entrega por lotes

Nombre en inglés: Batch turnaround time

Descripción: El tiempo que pasa entre el comienzo y el final de proceso de fondo.

Atributo al que ayuda a calcular: Comportamiento en el tiempo

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 561

Nombre en español: Capacidad de lotes

Nombre en inglés: Batch capacity

Descripción: Número de elementos de información que puede procesar de forma secuencial durante el proceso de fondo. Nota: Los elementos de información son, por ejemplo, solicitud, pagos, transferencias, etc.

Atributo al que ayuda a calcular: Comportamiento en el tiempo

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 562

Nombre en español: Tiempo de procesamiento

Nombre en inglés: Processing time

Descripción: El tiempo promedio y máximo que un usuario necesita para una tarea de transformación determinados, con una carga de uso determinado.

Nota: Este indicador es relevante cuando las tareas de procesamiento han de encajar con las acciones de los usuarios.

Atributo al que ayuda a calcular: Comportamiento en el tiempo

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 563

Nombre en español: Capacidad de procesamiento

Nombre en inglés: Processing capacity

Descripción: Número de tareas de procesamiento de un determinado tipo que el usuario puede realizar durante un período determinado con una carga de uso determinado. *Nota:* Este indicador es útil cuando una gran cantidad de tareas de procesamiento tienen que ser realizadas.

Atributo al que ayuda a calcular: Comportamiento en el tiempo

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 564

Nombre en español: Tiempo de transacción interna promedio

Nombre en inglés: Average internal transaction time

Descripción: Tiempo medio de una determinada tarea de procesamiento interno ocupa con una carga de uso seguro.

Atributo al que ayuda a calcular: Comportamiento en el tiempo

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 565

Nombre en español: Tiempo de transacción interna máximo

Nombre en inglés: Maximum internal transaction time

Descripción: Tiempo máximo de una determinada tarea de procesamiento interno ocupa con una carga de uso seguro.

Atributo al que ayuda a calcular: Comportamiento en el tiempo

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 566

Nombre en español: Tiempo transcurrido de la CPU

Nombre en inglés: CPU elapsed time

Descripción: Indica la velocidad de procesamiento mediante la medición del tiempo transcurrido entre el comienzo de la ejecución del programa y el final de la ejecución del programa.

Atributo al que ayuda a calcular: Comportamiento en el tiempo

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 567

Nombre en español: Tiempo de ejecución de la CPU

Nombre en inglés: CPU execution time

Descripción: Indica la velocidad de procesamiento mediante la medición del tiempo de CPU entre el comienzo de la ejecución del programa y el final de la ejecución del programa.

Atributo al que ayuda a calcular: Comportamiento en el tiempo

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 568

Nombre en español: Tiempo de procesamiento de E/S

Nombre en inglés: I/O processing time

Descripción: El tiempo requerido para la operación de E / entre la memoria principal y de almacenamiento externo.

Atributo al que ayuda a calcular: Comportamiento en el tiempo

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 569

Nombre en español: Tiempo de procesamiento de la red

Nombre en inglés: Network processing time

Descripción: El tiempo entre el comienzo de la transformación de terminal y el fin del procesamiento de la terminal.

Atributo al que ayuda a calcular: Comportamiento en el tiempo

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 570

Nombre en español: Tiempo de procesamiento del terminal

Nombre en inglés: Number of processed transactions

Descripción: Indica la capacidad del sistema de medición del número de transacciones realizadas en una unidad de tiempo.

Atributo al que ayuda a calcular: Comportamiento en el tiempo

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 572

Nombre en español: Número de transacciones procesadas

Nombre en inglés: Number of processed transactions

Descripción: Indica la capacidad del sistema de medición del número de transacciones realizadas en una unidad de tiempo.

Atributo al que ayuda a calcular: Comportamiento en el tiempo

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 572

Nombre en español: Ocupación de la comunicación

Nombre en inglés: Communication occupancy

Descripción: El número total de caracteres para ser transportados a través de una interfaz determinada con un número previsto de transacciones por período de tiempo. Nota: Este indicador es válido cuando la interfaz de comunicación es una interfaz privada. En el caso de una interfaz más pública, es más apropiado optar por algo distinto de 'número total de personajes. Una buena alternativa sería la de adaptar el indicador que se corresponden con la estructura de calificación de la comunicación pública de los medios. Cuando el número de caracteres por tipo de transporte no es constante, el número debe ser promediado

Atributo al que ayuda a calcular: Comportamiento de recursos

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 573

Nombre en español: Ocupación de la memoria interna

Nombre en inglés: Internal memory occupancy

Descripción: La cantidad de memoria interna necesaria para procesar una cierta cantidad de datos. Nota: La mayoría de sistemas operativos ofrecen funcionalidad para restringir la cantidad de memoria disponible o la medición de la cantidad máxima de memoria utilizada después de la ejecución de un programa

Atributo al que ayuda a calcular: Comportamiento de recursos

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 574

Nombre en español: Ocupación de la memoria externa

Nombre en inglés: External memory occupancy

Descripción: Volumen de memoria externa necesaria para procesar una cierta cantidad de datos. Nota: En general, la ocupación de memoria externa que aumenta linealmente con el número de elementos de datos.

Atributo al que ayuda a calcular: Comportamiento de recursos

Referencias: QUINT2 [58] [59]
Espectro: Oscuro

MÉTRICA 575

Nombre en español: Ocupación del procesador
Nombre en inglés: Processor occupancy
Descripción: Cantidad de tiempo utilizados. Nota: En general, un compromiso que debe hacerse entre los diferentes tipos de ocupación, a fin de lograr el "mejor" comportamiento de los recursos.
Atributo al que ayuda a calcular: Comprometimiento de recursos
Referencias: QUINT2 [58] [59]
Espectro: Oscuro

MÉTRICA 576

Nombre en español: Ocupación de la memoria real
Nombre en inglés: Real memory occupancy
Descripción: Indica la capacidad del sistema de medir el tamaño de memoria real para el programa.
Atributo al que ayuda a calcular: Comprometimiento de recursos
Referencias: QUINT2 [58] [59]
Espectro: Oscuro

MÉTRICA 577

Nombre en español: Ocupación de la memoria virtual
Nombre en inglés: Virtual memory occupancy
Descripción: indica la capacidad del sistema de medir el tamaño de la memoria virtual para el programa.
Atributo al que ayuda a calcular: Comprometimiento de recursos
Referencias: QUINT2 [58] [59]
Espectro: Oscuro

MÉTRICA 578

Nombre en español: Tamaño del WorkingSet
Nombre en inglés: Workingset size
Descripción: Indica la capacidad del sistema de medir el tamaño WorkingSet del programa
Atributo al que ayuda a calcular: Comprometimiento de recursos
Referencias: QUINT2 [58] [59]
Espectro: Oscuro

MÉTRICA 579

Nombre en español: Ocupación del archivo
Nombre en inglés: File occupancy
Descripción: Cantidad de archivos utilizados por el programa.

Atributo al que ayuda a calcular: Comprotamiento de recursos

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 580

Nombre en español: Ocupación de la red

Nombre en inglés: Network occupancy

Descripción: Cantidad de datos enviados y recibidos.

Atributo al que ayuda a calcular: Comprotamiento de recursos

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 581

Nombre en español: Utilización de CPU

Nombre en inglés: CPU utilisation

Descripción: Relación entre la utilización de la CPU principal de la unidad de tiempo adecuado.

Atributo al que ayuda a calcular: Comprotamiento de recursos

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 582

Nombre en español: Utilización de memoria principal

Nombre en inglés: Main memory utilisation

Descripción: Relación entre la utilización de la memoria principal en la unidad de tiempo adecuado.

Atributo al que ayuda a calcular: Comprotamiento de recursos

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 583

Nombre en español: Utilización de canal de E/S

Nombre en inglés: I/O channel utilisation

Descripción: proporción de uso de E / S de tiempo de canal utilizado en la unidad de tiempo adecuado.

Atributo al que ayuda a calcular: Comprotamiento de recursos

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 584

Nombre en español: Utilización de archivo

Nombre en inglés: File utilisation

Descripción: Relación entre la utilización de archivos en la unidad de tiempo adecuado.

Atributo al que ayuda a calcular: Comprotamiento de recursos

Referencias: QUINT2 [58] [59]
Espectro: Oscuro

MÉTRICA 585

Nombre en español: Utilización de red
Nombre en inglés: Network utilisation
Descripción: Relación entre la utilización de la red en la unidad de tiempo adecuado
Atributo al que ayuda a calcular: Comprometimiento de recursos
Referencias: QUINT2 [58] [59]
Espectro: Oscuro

MÉTRICA 586

Nombre en español: Utilización de dispositivo de E/S
Nombre en inglés: I/O device utilisation
Descripción: Proporción de I / O de utilización del dispositivo en la unidad de tiempo adecuado
Atributo al que ayuda a calcular: Comprometimiento de recursos
Referencias: QUINT2 [58] [59]
Espectro: Oscuro

MÉTRICA 587

Nombre en español: Utilización de terminal
Nombre en inglés: Terminal utilisation
Descripción: Relación entre la utilización de la terminal en la unidad de tiempo adecuado.
Atributo al que ayuda a calcular: Comprometimiento de recursos
Referencias: QUINT2 [58] [59]
Espectro: Oscuro

MÉTRICA 588

Nombre en español: Proporción de reconocimiento de posición de fallas
Nombre en inglés: Fault position recognition ratio
Descripción: La relación entre el número de fallos que los usuarios reconocen correctamente las posiciones del fallo en el número de fallos detectados causados por las fallas del software, como consecuencia del mantenedor de analizar los fracasos.
Atributo al que ayuda a calcular: Analizabilidad
Referencias: QUINT2 [58] [59]
Espectro: Oscuro

MÉTRICA 589

Nombre en español: Tiempo medio de análisis de falla
Nombre en inglés: Mean failure analysis time

Descripción: El tiempo medio necesario para analizar un fracaso, y para descubrir los defectos derivados de este fracaso, y por separado las posiciones que ser reparado por el encargado, quien recibió el informe el fracaso.

Atributo al que ayuda a calcular: Analizabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 590

Nombre en español: Esfuerzo de modificación por unidad de volumen

Nombre en inglés: Modification effort per unit volume

Descripción: Cantidad media de esfuerzo necesario para modificar el producto de software, por unidad de volumen de la modificación. Nota: El protocolo de medición exige un esfuerzo considerable cuando los cambios de ficción, se proponen poner a prueba la mutabilidad.

Atributo al que ayuda a calcular: Cambiabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 591

Nombre en español: Esfuerzo de corrección por defecto

Nombre en inglés: Correction effort per defect

Descripción: Significa esfuerzo necesario para reparar un defecto en el producto de software. Nota: Este indicador sólo puede medirse al probar el producto de software. Sin embargo, en primer lugar el desarrollo de una parte representativa del producto, los efectos de las decisiones de diseño se pueden medir tempranos. La objetividad es influenciada por la posibilidad de dividir el defecto en los defectos más pequeños, reduciendo así el esfuerzo de corrección por defecto. Criterios para el tamaño de un defecto que se determine.

Atributo al que ayuda a calcular: Cambiabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 592

Nombre en español: Tiempo medio de corrección de falla

Nombre en inglés: Mean fault correction time

Descripción: El tiempo medio desde la recepción del informe no se envía el software de corrección en la corrección de fallos.

Atributo al que ayuda a calcular: Cambiabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 593

Nombre en español: Tiempo medio de tratamiento de falla

Nombre en inglés: Mean failure treatment time

Descripción: La media del tiempo de producido el fracaso de la restauración para los usuarios finales. Nota: el fracaso del tratamiento mediante la resolución del problema que incluye la búsqueda y corrección de errores derivados de fallos, o para proporcionar una operación de procedimientos alternativos para evitar los fracasos.

Atributo al que ayuda a calcular: Cambiabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 594

Nombre en español: Tiempo medio de trabajo para corrección de falla

Nombre en inglés: Mean fault correction work time

Descripción: El tiempo medio de trabajo para corregir los errores de software descubiertos que han de corregirse en la corrección de fallas, como consecuencia del análisis de la falla. Nota: Esta es una versión simplificada de corrección de Esfuerzo por defecto.

Atributo al que ayuda a calcular: Cambiabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 595

Nombre en español: Tiempo de trabajo de revisión significativa por líneas de código cambiadas en el código fuente

Nombre en inglés: Mean revise work time per changed line of source code

Descripción: El tiempo medio de trabajo para la revisión y actualización por añadido línea de código fuente. Nota: Esta es una versión simplificada de modificación esfuerzo por unidad de volumen.

Atributo al que ayuda a calcular: Cambiabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 596

Nombre en español: Grado de cumplimiento del documento de Mantenimiento

Nombre en inglés: Fulfilment degree of maintenance document

Descripción: Grado de cumplimiento de documentos de los productos que son utilizables en la etapa de Mantenimiento. Nota: Esto se puede expresar en una relación del número de páginas de documentos para el Mantenimiento dividido por el número de líneas de código fuente.

Atributo al que ayuda a calcular: Cambiabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 597

Nombre en español: Proporción de nuevas fallas en la revisión

Nombre en inglés: Ratio of new faults at revisión

Descripción: La relación de nuevas faltas hechas en la revisión.

Atributo al que ayuda a calcular: Estabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 598

Nombre en español: Esfuerzo de prueba por unidad de volumen

Nombre en inglés: Test effort per unit volume

Descripción: Esfuerzo necesario para poner a prueba una unidad de volumen del producto de software con un cierto grado de cobertura de las pruebas. Nota: La validez de este indicador depende de la experiencia del equipo de pruebas y los equipos del entorno de prueba. Este indicador sólo puede medirse después de la finalización del producto de software. Sin embargo, una estimación se puede hacer después de terminar una parte representativa del producto.

Atributo al que ayuda a calcular: Testeabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 599

Nombre en español: Número de casos de prueba por unidad de volumen

Nombre en inglés: Number of test cases per unit volume

Descripción: Número de casos de prueba que deben hacerse para poner a prueba una unidad de volumen del producto de software con un cierto grado de cobertura de las pruebas.

Atributo al que ayuda a calcular: Testeabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 600

Nombre en español: promedio de tiempo de trabajo del usuario para verificar la corrección de error

Nombre en inglés: Mean user's work time to verify the fault correction

Descripción: verificar la corrección de error de La media de tiempo de trabajo del usuario para verificar la corrección de fallos.

Atributo al que ayuda a calcular: Testeabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 601

Nombre en español: Tiempo de trabajo medio para la prueba de corrección de fallos

Nombre en inglés: Mean work time to test the fault correction

Descripción: El tiempo medio de trabajo mantenedor para probar la corrección de error, después de corregir la falla.

Atributo al que ayuda a calcular: Testeabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 602

Nombre en español: Tiempo de prueba para líneas de código cambiadas

Nombre en inglés: Test work time per changed source code line

Descripción: Tiempo de trabajo para poner a prueba actualizado y añadido el código fuente por unidad de línea en caso de corrección de fallas y / o el caso de remodelación.

Atributo al que ayuda a calcular: Testeabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 603

Nombre en español: Proporción de control de esfuerzo

Nombre en inglés: Control effort ratio

Descripción: La relación de esfuerzo puesto en el control del producto de software (incluido el Mantenimiento) en horas-hombre para el número de horas que el producto está disponible para los usuarios. Nota: el esfuerzo de control pueden estar escondidos en las acciones del usuario, por lo tanto influyen en este indicador.

Atributo al que ayuda a calcular: Manejabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 605

Nombre en español: Proporción de partes rehusadas

Nombre en inglés: Ratio reused parts

Descripción: Coeficiente de Reutilización de partes del producto de software para el número total de partes del producto de software. Nota: Este indicador no tiene en cuenta el número de veces que ciertas partes son reutilizados. La validez de este indicador depende también de la "Reutilización de manejo" de la organización. La proporción de partes reutilizadas sólo pueden medirse después de la liberación del producto de software.

Atributo al que ayuda a calcular: Manejabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 606

Nombre en español: Esfuerzo de Portabilidad

Nombre en inglés: Effort for portability

Descripción: Significa esfuerzo necesario para adaptar una unidad de volumen del producto de software a una plataforma especifica diferentes. Nota: Este indicador sólo puede medirse por el ejercicio efectivo de la adaptación.

Atributo al que ayuda a calcular: Adaptabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 607

Nombre en español: Coeficiente aplicable del entorno de hardware

Nombre en inglés: Applicable ratio of hardware environment

Descripción: Coeficiente aplicable del entorno de hardware sin tener que cambiar el software. Nota: El número de máquinas de aplicación se divide por el número de máquinas de aplicación que se establezcan.

Atributo al que ayuda a calcular: Adaptabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 608

Nombre en español: Coeficiente aplicable del entorno de sistema operativo

Nombre en inglés: Applicable ratio of OS environment

Descripción: Coeficiente aplicable del entorno de sistema operativo sin modificar el software. Nota: El número de OS aplicable se divide por el número de OS aplicable a establecerse.

Atributo al que ayuda a calcular: Adaptabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 609

Nombre en español: Coeficiente aplicable de medio ambiente de datos

Nombre en inglés: Applicable ratio of data environment

Descripción: Coeficiente aplicable del entorno de datos sin cambiar el software. Nota: El número de datos aplicables, se divide por el número de datos aplicables que se establezcan.

Atributo al que ayuda a calcular: Adaptabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 610

Nombre en español: Coeficiente aplicable de medio ambiente de operación

Nombre en inglés: Applicable ratio of operation environment

Descripción: Coeficiente aplicable del manual de funcionamiento y el procedimiento, sin cambiar el software. Nota: El número de manuales y procedimientos aplicables, se divide por el número de manuales y procedimientos aplicables que se establezcan.

Atributo al que ayuda a calcular: Adaptabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 611

Nombre en español: Proporción de cambio de parámetro

Nombre en inglés: Parameter change ratio

Descripción: Relación de corrección de parámetros que debe ser cambiado mediante la transferencia del software.

Atributo al que ayuda a calcular: Instalabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 612

Nombre en español: Proporción de recopilación del programa

Nombre en inglés: Recompile program ratio

Descripción: Programa de relación que se va a volver a compilar mediante la transferencia del software.

Atributo al que ayuda a calcular: Instalabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 613

Nombre en español: Proporción de cambio de archivo

Nombre en inglés: File change ratio

Descripción: Archivo relación de corrección que se va a cambiar por la transferencia del software.

Atributo al que ayuda a calcular: Instalabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 614

Nombre en español: Proporción de cambio de lista de salida

Nombre en inglés: Output list change ratio

Descripción: Lista de la relación producto de corrección que se va a cambiar por la transferencia del software. Nota: Los indicadores 612 a 615 sólo Aplican cuando la instalación incluye la (re) compilar.

Atributo al que ayuda a calcular: Instalabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 615

Nombre en español: Proporción de Conformidad del estándar

Nombre en inglés: Standard conformance ratio

Descripción: Conformidad relación en la que el software está en Conformidad con las leyes, disposiciones y convenios. Nota: Esta relación es difícil de determinar. Facilidad de uso de este indicador es cuestionable.

Atributo al que ayuda a calcular: Conformidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 616

Nombre en español: Proporción de cambio de función

Nombre en inglés: Function change ratio

Descripción: Relación de funciones para ser cambiado por la transferencia del software.

Atributo al que ayuda a calcular: Remplazabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

MÉTRICA 617

Nombre en español: Proporción de cambio de código fuente

Nombre en inglés: Source code change ratio

Descripción: Relación de código fuente para ser cambiado por la transferencia del software.

Atributo al que ayuda a calcular: Remplazabilidad

Referencias: QUINT2 [58] [59]

Espectro: Oscuro

ANEXO I: LAS HEURÍSTICAS DE CALIDAD ENCONTRADAS

A continuación son mostradas las heurísticas encontradas, organizadas de la siguiente manera:

Heurística <Nº>
<Nombre>
<Descripción>
<Disciplina>
<Atributos relacionados>
<Referencia>

En seguida esta la lista de la totalidad de métricas de calidad de software encontradas.

HEURÍSTICA 001

Nombre: Heurística basada en NOS (Número de pasos)

Descripción:

Los casos de uso demasiado cortos suelen ser incompletos o no representan realmente una interacción actor–sistema resultando generalmente triviales. Por otro lado, los casos de uso demasiado largos son frecuentemente incomprensibles, presentan un nivel de detalle excesivo [Lilly 1999] o avanzan a un ritmo muy lento, es decir, son dirigidas con demasiada lentitud a alcanzar el objetivo del caso de uso siendo más difíciles de leer.

Rango recomendado: [4,9].

Disciplina: Análisis.

Atributos relacionados: Comprensibilidad, Documentación, Testeabilidad.

Referencia: BERNÁRDEZ [81]

HEURÍSTICA 002

Nombre: Heurística basada en NOAS/NOS (proporción de pasos por actor)

Descripción:

Esta heurística esta basada en la idea de que un caso de uso sirve básicamente para expresar una interacción actor–sistema. Por ello, el número de pasos de actor y el de pasos de sistema deben estar en torno al 50%, considerando también la posibilidad de que existan pasos de inclusión o extensión que los realice otro caso de uso.

Rango recomendado: [30%,60%]

Disciplina: Análisis.

Atributos relacionados: Comprensibilidad, Documentación, Testeabilidad.

Referencia: BERNÁRDEZ [81]

HEURÍSTICA 003

Nombre: Heurística basada en NOSS/NOS (proporción de pasos de sistema)

Descripción:

En situaciones con NOSS/NOS alto el defecto más habitual es que pretende representar un proceso bacht que no necesita participación de ningún actor, por lo que no es realmente un caso de uso. En estos casos lo más conveniente es

expresar el requisito mediante un requisito funcional redactado en texto plano, sin usar la estructura de caso de uso.

Algunos casos de uso abstractos con NOSS/NOS por debajo del 40% no presentan defectos porque, al ser un caso de uso abstracto un fragmento de otros casos de uso, puede ocurrir que los pasos de actor estén descritos en los caso de uso que lo activan. Si el caso de uso es concreto y pretende representar una interacción actor–sistema, debe incluirse al menos un paso de actor, el primero, algo que no tiene porqué ser cierto en los casos de uso abstractos.

Rango recomendado: [40%,80%]

Disciplina: Análisis.

Atributos relacionados: Comprensibilidad, Documentación, Testeabilidad.

Referencia: BERNÁRDEZ [81]

HEURÍSTICA 004

Nombre: Heurística basada en NOUS/NOS (proporción de pasos de casos de uso)

Descripción:

Esta heurística esta basada en la idea de que los mecanismos de estructuración de los casos de uso conocidos como inclusión y extensión deben usarse principalmente como medio para evitar repeticiones de secuencias de pasos en distintos casos de uso:

- Los principales defectos que predice la heurística son: La introducción de aspectos de navegación en los casos de uso, de forma que crea una especie de caso de uso menú principal que es extendido por otros casos de uso en función de la elección del actor principal.
- La tendencia a organizar los casos de uso como si fueran un programa de ordenador, intentando modularizar en exceso la especificación, con las negativas repercusiones que ello acarrea para los clientes y usuarios no familiarizados con la programación.

Rango recomendado: [0%,35%]

Disciplina: Análisis.

Atributos relacionados: Comprensibilidad, Documentación, Testeabilidad.

Referencia: BERNÁRDEZ [81]

HEURÍSTICA 005

Nombre: Heurística basada en CC (Complejidad ciclomatica)

Descripción:

La idea en la que esta basada esta heurística es que los casos de uso con abundancia de caminos alternativos suelen ser difíciles de entender y, a veces, hasta ilegibles. Para conocer el número de caminos alternativos definiendo la Complejidad ciclomatica (CC) de un caso de uso de la misma forma que define la CC de McCabe [92] para código fuente: el número de puntos de decisión más 1, es decir, el número de pasos condicionales más el número de excepciones más 1.

Rango recomendado: [1,5]

Disciplina: Análisis.

Atributos relacionados: Comprensibilidad, Documentación, Testeabilidad.

Referencia: BERNÁRDEZ [81]

HEURÍSTICA 006

Nombre: Chequear el número de pasos

Descripción:

Desde que los casos de uso describen interacciones entre actores y el sistema, un caso de uso debería tener al menos 2 pasos para una interacción mínima actor requiere – sistema responde. Los casos de uso con NOS < 2 deberían ser chequeados por no-completitud potencial. De otro lado, un valor muy alto de NOS (por ejemplo 20) es un indicador de excesiva Complejidad o de la carencia de estructura (i.e. relaciones include o extend) en casos de uso.

Disciplina: Análisis.

Atributos relacionados: Eficiencia, Comprensibilidad, Documentación, Testeabilidad.

Referencia: DURAN [98]

HEURÍSTICA 007

Nombre: Chequear la tasa de diferentes tipos de pasos

Descripción:

Los tres tipos de pasos en los casos de uso en REM, paso acción de actor, paso acción de sistema y paso acción de caso de uso, deberían no aparecer con la misma frecuencia. Usando el mismo razonamiento que en la heurística previa, desde que los casos de uso describen interacciones entre actores y el sistema, aproximadamente la mitad de los pasos de un caso de uso deberían ser pasos acción actor y acción sistema. De otro lado, un caso de uso no debería tener de sus pasos siendo inclusiones o extensiones. En otras palabras, un valor alto de NOUS es un indicador claro de un uso abusivo de relaciones de casos de uso.

Pensamos que los siguientes son tasas razonables para los tres tipos diferentes de pasos:

- NOAS/NOS 2 [30%,60%]

- NOSS/NOS 2 [30%,60%]

- NOUS/NOS 2 [0%,40%]

Disciplina: Análisis.

Atributos relacionados: Mantenibilidad

Referencia: DURAN [98]

HEURÍSTICA 008

Nombre: Chequear el número de excepciones

Descripción:

Si un caso de uso no tiene excepciones o solo unas pocas comparado con el número de pasos, es probablemente porque solamente han sido definidas secuencias ordinarias de pasos y debería ser chequeado para completitud. De lo contrario, un caso de uso con la mayoría de estos casos que tienen asociadas excepciones probablemente es muy complejo para ser entendido. La forma de la situación puede ser detectada buscando caso de uso con NOE/NOS inferior a 5%, en tanto que el superior puede ser detectado buscando casos de uso con NOE/NOS sobre 50%.

Disciplina: Análisis.

Atributos relacionados: Mantenibilidad.

Referencia: DURAN [98]

HEURÍSTICA 009

Nombre: Chequear la Complejidad \square iclomática

Descripción:

A pesar del hecho que los casos de uso son muy diferentes desde su código fuente, la heurística de conservación de la Complejidad \square iclomática (CC) por debajo de 10 puede también ser aplicado a los casos de uso.

La Complejidad \square iclomática para casos de uso en REM, puede ser calculada como el número de puntos de decisión + 1, i.e. el número de pasos condicionales + el número de excepciones + 1.

Disciplina: Análisis.

Atributos relacionados: Mantenibilidad, Reusabilidad.

Referencia: DURAN [98]

HEURÍSTICA 010

Nombre: Encontrar objetos del mundo real

Descripción:

La primer y más importante aproximación para identificar las alternativas de diseño en una aproximación "a partir del libro" orientado a objetos, con un enfoque sobre la identificación de objetos sintéticos y del mundo real. Los pasos para diseñar con objetos son:

- Identificar los objetos y sus atributos (métodos y datos)
- Determinar que puede ser realizado por cada objeto.
- Determinar las partes de cada objeto que será visible a otros objetos---qué partes serán públicas y privadas.
- Definir la interfaz pública de cada objeto.

Disciplina: Análisis.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: MCCONNELL [66]

HEURÍSTICA 011

Nombre: Abstracciones consistentes con la forma

Descripción:

La Abstracción es la capacidad de interconectar con un concepto mientras seguramente ignora algunos de sus detalles- manejando diferentes detalles a diferentes niveles. Las clases base son abstracciones que permiten enfocarse sobre atributos comunes de un conjunto de clases derivadas e ignora los detalles de clases específicas mientras está trabajando sobre la clase base. Una buena clase interfaz es una Abstracción que permite enfocarse sobre la interfaz sin necesidad de preocuparse sobre el trabajo interno de la clase. La interfaz para una rutina bien diseñada provee el mismo beneficio a un bajo nivel de detalle, y la interfaz para un paquete bien diseñado o subsistema provee dicho beneficio a un alto nivel de detalle. Desde un punto de vista de la Complejidad, el principal beneficio de la Abstracción es que le permite ignorar detalles irrelevantes

Disciplina: Análisis, Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: MCCONNELL [66]

HEURÍSTICA 012

Nombre: Encapsular los detalles de Implementación.

Descripción:

La encapsulación recoge lo que la Abstracción deja. La Abstracción dice, “le está permitido buscar en un objeto a alto nivel de detalle”. La encapsulación dice, “por eso, no permite buscar en un objeto a cualquier nivel de detalle”.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: MCCONNELL [66]

HEURÍSTICA 013

Nombre: Heredar cuando la jerarquía simplifique el diseño.

Descripción:

Definiendo similitudes y diferencias entre objetos es llamado “Herencia” porque la parte de los empleados de tiempo parcial y tiempo completo hereda características del tipo empleado general.

La Herencia es uno de las herramientas más poderosas. Puede dar grandes beneficios cuando sea usado bien y puede hacer bastante daño cuando sea usado ingenuamente.

Disciplina: Diseño.

Atributos relacionados:

Referencia: MCCONNELL [66]

HEURÍSTICA 014

Nombre: Ocultar secretos (ocultamiento de Información).

Descripción:

El ocultamiento de información es parte de fundamentos de tanto el diseño Estructurado como el diseño orientado a objetos. En el diseño Estructurado, la noción de “cajas negras” viene del ocultamiento de la información. En el diseño orientado a objetos, este da pie a conceptos de encapsulación y modularidad, y está asociado con el concepto de Abstracción.

Disciplina: Análisis, Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: MCCONNELL [66]

HEURÍSTICA 015

Nombre: Secretos y el derecho a la privacidad.

Descripción:

Una tarea clave en el diseño de una clase es decidir cuáles características deberían ser conocidas fuera de la clase y cuáles de las restantes en secreto. Una clase debería usar 25 rutinas y exponer solamente 5 de ellas, usando las otras 20 internamente. Una clase debería usar varios tipos de datos y exponer ninguna información acerca de ellos. Este aspecto del diseño de clases es conocido como “visibilidad” desde lo que esta tiene que hacer con aquellas características de la clase que están “visibles” o “expuestas” al exterior de la clase.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: MCCONNELL [66]

HEURÍSTICA 016

Nombre: Barreras para el ocultamiento de información

Descripción:

En un conjunto de instancias, el ocultamiento de información es verdaderamente imposible, pero muchas de esas barreras de información son bloques mentales contruidos sobre el uso habitual de otras técnicas.

Distribución excesiva de información.

Dependencias Circulares.

Datos de clase equivocados para datos globales.

Penalizaciones de Desempeño percibidas.

Disciplina: Análisis, Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: MCCONNELL [66]

HEURÍSTICA 017

Nombre: El valor del ocultamiento de la información.

Descripción:

El ocultamiento de la información tiene un poder heurístico único, una capacidad única para inspirar soluciones de diseño efectivo. El diseño tradicional orientado a objetos provee un poder heurístico único para modelar el mundo en objetos, pero el pensamiento en objetos no le ayudaría a eliminar la declaración de un ID como un entero en lugar de un IdType.

Disciplina: Análisis, Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: MCCONNELL [66]

HEURÍSTICA 018

Nombre: Identifique áreas probables para cambiar

Descripción:

He aquí unas áreas probables a cambiar:

Lógica de negocio: estas tienden a ser el origen de frecuentes cambios en el software. De aquí la importancia del ocultamiento de la información, dado que la lógica permanecerá oculta mientras los cambios no sean realmente necesarios.

Dependencias de hardware: esto ayuda cuando es desarrollado software para hardware volátil, es decir, aquél que constantemente de esta conectando o desconectando y/o cambiando de lugar.

Entrada y salida: La entrada y salida es un área volátil. En general, es una buena idea examinar todas las interfaces externas para posibles cambios.

Características no estándar del lenguaje: si usa extensiones no estándar en su lenguaje de programación, oculte esas extensiones en una clase de las suyas tal que pueda reemplazarlos con su propio código cuando lo mueva a un ambiente diferente. Así mismo, si usa rutinas de librerías que no están disponibles en todos los ambientes, oculte las rutinas de las librerías tras una interfaz que trabaje también en algún otro ambiente.

Dificultades en áreas de diseño y construcción: es buena idea ocultarlas porque deben estar hechas pobremente y usted debería rehacerlas.

Variables de estado: las variables de estado indican es estado de un programa y tienden a ser cambiados más frecuentemente que la mayoría de otros datos. En un escenario típico, debería definir una variable de estado del error como una

variable bool y decidir luego qué debería ser implementada como un tipo enumerado con los valores respectivos.

Restricciones del tamaño de datos: Cuando declara un arreglo de tamaño 15, está exponiendo información al mundo que el mundo no necesita conocer. ¡Defienda su derecho a la privacidad! El ocultamiento de la información no es siempre tan complicado como una clase amplia.

Disciplina: Mantenimiento.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: MCCONNELL [66]

HEURÍSTICA 019

Nombre: Anticipar diferentes grados de cambio.

Descripción:

Una buena técnica para identificar áreas probables a cambiar es primero identificar el subconjunto mínimo del programa que debería ser usado por el usuario. El subconjunto constituye el núcleo del sistema y es poco probable de cambio. Pueden ser bastante pequeños que parezcan triviales. Esas áreas de mejora potencial constituyen cambios potenciales en el sistema; diseñe dichas áreas usando los principios de ocultamiento de la información.

Disciplina: Mantenimiento.

Atributos relacionados: Mantenibilidad, Reusabilidad.

Referencia: MCCONNELL [66]

HEURÍSTICA 020

Nombre: Conserve bajo acoplamiento

Descripción:

El acoplamiento describe que tan fuerte un módulo (clase o rutina) está relacionado con otros.

- criterios de acoplamiento
- tipos de acoplamiento.

Disciplina: Análisis, Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad.

Referencia: MCCONNELL [66]

HEURÍSTICA 021

Nombre: Apunte hacia fuerte cohesión

Descripción:

Las clases que contienen Funcionalidad fuertemente relacionada están descritas como que tienen fuerte cohesión, y la meta de la heurística es hacer la cohesión tan fuerte como sea posible. La cohesión es una herramienta útil para el manejo de la Complejidad porque la mayoría de código en una clase soporta un propósito general, la manera más fácil en que su mente puede recordar cada cosa que hace el código.

Disciplina: Análisis, Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad.

Referencia: MCCONNELL [66]

HEURÍSTICA 022

Nombre: Construir jerarquías

Descripción:

Las jerarquías son herramientas usables para lograr el imperativo técnico primario del software porque le permiten enfocarse solamente sobre el nivel de detalle que le concierne actualmente. Los detalles no son alejados completamente, simplemente son puestos en otro nivel así que puede pensar en ellos cuando quiera en lugar de pensar respecto a los detalles todo el tiempo.

Disciplina: Análisis, Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad.

Referencia: MCCONNELL [66]

HEURÍSTICA 023

Nombre: Formalizar contratos de clase.

Descripción:

A un nivel más detallado, pensar de cada interfaz de clase como un contrato con el resto del programa puede conllevar buenas ideas. Típicamente, el contrato es algo como “si promete darme los datos X, Y y Z y usted promete que tendrán las características A, B y C, prometo desarrollar las operaciones 1,2 y 3 con las restricciones 8,9 y10”.... Los contratos son usables para el manejo de la Complejidad porque, al menos en teoría, el objeto puede seguramente ignorar cualquier comportamiento no contractual. En la práctica, este aspecto es mucho más difícil.

Disciplina: Diseño.

Atributos relacionados: Corrección, Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: MCCONNELL [66]

HEURÍSTICA 024

Nombre: Asignar Responsabilidades.

Descripción:

Otra heurística para pensar a través de cómo deberían ser asignadas las responsabilidades a objetos. Preguntando para qué debería ser responsable cada objeto es similar a preguntar qué información debería ocultar.

Disciplina: Diseño, Implementación.

Atributos relacionados: Mantenibilidad, Reusabilidad.

Referencia: MCCONNELL [66]

HEURÍSTICA 025

Nombre: Diseño por pruebas

Descripción:

Un raciocinio que puede generar ideas de diseños interesantes es preguntar el sistema a qué se parecerá si lo diseña para facilitar pruebas. ¿Necesita separar la interfaz de usuario del resto del código para que pueda ejercitar independientemente? ¿Necesita organizar cada subsistema tal que minimice dependencias de otros subsistemas? El diseño para pruebas tiende a resultar en interfaces de clase más formalizadas, que generalmente es benéfico.

Disciplina: Diseño.

Atributos relacionados: Eficiencia, Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: MCCONNELL [66]

HEURÍSTICA 026

Nombre: Elimine fallas.

Descripción:

Muchas de las fallas de puentes más espectaculares han ocurrido porque son enfocados en éxitos previos y no consideran adecuadamente modos de posibles fallas. Algunas fallas, como la del puente de Tacoma Narrows, podría haber sido evitadas si los diseñadores hubiesen considerado cuidadosamente las posibilidades de que el puente pudiera fallar y no solo copiar los atributos de otros diseños exitosos.

Disciplina: Análisis, Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: MCCONNELL [66]

HEURÍSTICA 027

Nombre: Escoja el tiempo de enlace concienzudamente.

Descripción:

El tiempo de enlace refiere al tiempo que un valor específico es enlazado a una variable. El código que enlaza temprano tiende a ser simple, pero también tiende a ser menos flexible.

Disciplina: Diseño, Implementación.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: MCCONNELL [66]

HEURÍSTICA 028

Nombre: Elaborar puntos centrales de control

Descripción:

El control puede estar centralizado en clases, rutinas, macros de preprocesador, archivos de `#include`- aún una constante mencionada es una ejemplo de punto de control central.

El beneficio de la Complejidad reducida esta en los pocos lugares en que tiene que buscar algo; la Facilidad y seguridad cambiarán.

Disciplina: Implementación.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: MCCONNELL [66]

HEURÍSTICA 029

Nombre: Considere el uso de la fuerza bruta.

Descripción:

Una herramienta heurística poderosa es la fuerza bruta. No la subestime. Una solución por fuerza bruta que trabaja es mejor que una solución elegante que no trabaja. Puede tomar un largo tiempo tener una solución elegante que trabaje.

Disciplina: Diseño, Implementación.

Atributos relacionados: Eficiencia, Mantenibilidad.

Referencia: MCCONNELL [66]

HEURÍSTICA 030

Nombre: Dibuje un diagrama

Descripción:

Los diagramas son otra herramienta heurística poderosa. Una imagen vale más que 1000 palabras. Usted actualmente busca dejar más de 1000 palabras porque un punto de uso de una imagen es que ella puede representar el problema con alto nivel de Abstracción.

Disciplina: Análisis, Diseño.

Atributos relacionados: Testeabilidad.

Referencia: MCCONNELL [66]

HEURÍSTICA 031

Nombre: Conserve su diseño modular.

Descripción:

La meta de la modularidad es hacer cada rutina o clase como una “caja negra”. Usted sabe qué ingresa, y sabe qué sale, pero no sabe qué sucede al interior. Una caja negra tiene tanto una interfaz simple como una Funcionalidad bien definida que para alguna entrada específica puede predecir adecuadamente la salida correspondiente. Si sus rutinas son como cajas negras, son perfectamente modulares, desarrollan acciones bien definidas y tienen interfaces simples. El concepto de modularidad esta relacionado con el ocultamiento de información, Encapsulamiento y otras heurísticas de diseño. Pero algunas veces pensando en cómo ensamblar un sistema desde un conjunto de cajas negras da ideas que el ocultamiento de información y el Encapsulamiento no dan, así que su valor lo tiene en su bolsillo de atrás.

Disciplina: Diseño.

Atributos relacionados: Portabilidad, Corrección, Eficiencia, Mantenibilidad, Reusabilidad, Testeabilidad

Referencia: MCCONNELL [66]

HEURÍSTICA 032

Nombre: Heurística 2.1

Descripción:

Todos los datos deberían estar ocultos en su clase.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 033

Nombre: Heurística 2.2

Descripción:

Los usuarios de una clase deben ser dependientes sobre su interfaz pública, pero una clase no debería ser dependiente de sus usuarios.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 034

Nombre: Heurística 2.3

Descripción:

Minimice el número de mensajes en el protocolo de una clase.

Disciplina: Diseño.

Atributos relacionados: Eficiencia, Mantenibilidad, Reusabilidad.

Referencia: Riel [82]

HEURÍSTICA 035

Nombre: Heurística 2.4

Descripción:

Implemente una interfaz pública mínima que todas las clases entiendan [e.g. operaciones tales como copiar (profundo versus superficial), prueba de igualdad, impresión bonita, conversión desde una descripción ASCII, etc.]

Disciplina: Diseño.

Atributos relacionados: Flexibilidad, Eficiencia, Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 036

Nombre: Heurística 2.5

Descripción:

No coloque detalles de implementación tales como funciones privadas de código compartido en la interfaz pública de una clase.

Disciplina: Diseño, Implementación.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 037

Nombre: Heurística 2.6

Descripción:

No desordene la interfaz pública de una clase con cosas que los usuarios de la clase no son capaces de usar o no están interesados en usar.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Corrección

Referencia: Riel [82]

HEURÍSTICA 038

Nombre: Heurística 2.7

Descripción:

Las clases deberían solo exhibir nulo o exportar acoplamiento con otras clases, esto es, una clase debería solo usar operaciones en la interfaz pública de otra clase o tener nada que hacer con esa clase.

Disciplina: Diseño.

Atributos relacionados: Eficiencia, Mantenibilidad, Reusabilidad, Testeabilidad

Referencia: Riel [82]

HEURÍSTICA 039

Nombre: Heurística 2.8

Descripción:

Una clase debería capturar una y sólo una Abstracción clave.

Disciplina: Análisis, Diseño.

Atributos relacionados: Corrección, Mantenibilidad, Reusabilidad, Testeabilidad

Referencia: Riel [82]

HEURÍSTICA 040

Nombre: Heurística 2.9

Descripción:

Conserve relacionados datos y comportamiento en un lugar.

Disciplina: Diseño.

Atributos relacionados: Corrección, Mantenibilidad, Reusabilidad, Testeabilidad

Referencia: Riel [82]

HEURÍSTICA 041

Nombre: Heurística 2.10

Descripción:

Derive información no relacionada en otra clase (i.e. comportamiento sin comunicación).

Disciplina: Diseño.

Atributos relacionados: Reusabilidad.

Referencia: Riel [82]

HEURÍSTICA 042

Nombre: Heurística 2.11

Descripción:

Asegúrese que las abstracciones de su modelo son clases y no simplemente roles que juegan los objetos.

Disciplina: Diseño.

Atributos relacionados: Corrección, Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 043

Nombre: Heurística 3.1

Descripción:

Distribuya la inteligencia del sistema horizontalmente como uniformemente sea posible, esto es, las clases del nivel superior en un diseño deberían compartir el trabajo uniformemente.

Disciplina: Diseño, Implementación.

Atributos relacionados: Mantenibilidad

Referencia: Riel [82]

HEURÍSTICA 044

Nombre: Heurística 3.2

Descripción:

No cree clases u objetos Dios en su sistema. Sospeche de una clase cuyo nombre contenga Manejador, Director, Sistema, o Subsistema.

Disciplina: Diseño, Implementación.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad. *Referencia:* Riel [82]

HEURÍSTICA 045

Nombre: Heurística 3.3

Descripción:

Tenga cuidado de clases que tienen muchos métodos de acceso definidos en su interfaz pública. Tener mucho implica que los datos relacionados y el comportamiento no están siendo conservados en un lugar.

Disciplina: Diseño.

Atributos relacionados: Corrección, Interoperabilidad, Mantenibilidad, Reusabilidad, Testeabilidad

Referencia: Riel [82]

HEURÍSTICA 046

Nombre: Heurística 3.4

Descripción:

Tenga cuidado de clases que tienen mucho comportamiento sin comunicación, esto es, métodos que operan sobre un subconjunto propio de datos miembro de una clase. Las clases Dios a menudo exhiben gran trato de comportamiento sin comunicación.

Disciplina: Diseño.

Atributos relacionados: Corrección, Interoperabilidad, Mantenibilidad, Reusabilidad, Testeabilidad

Referencia: Riel [82]

HEURÍSTICA 047

Nombre: Heurística 3.5

Descripción:

En aplicaciones que constan de un modelo interactivo orientado a objetos con una interfaz de usuario, el modelo debería nunca ser dependiente sobre la interfaz. La interfaz debería ser dependiente del modelo.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Corrección, Reusabilidad.

Referencia: Riel [82]

HEURÍSTICA 048

Nombre: Heurística 3.6

Descripción:

Modele el mundo real toda vez que sea posible. (Esta heurística es a menudo violada por razones de distribución de la inteligencia del sistema, eliminación de clases Dios, y conservación de datos y comportamiento relacionados en un lugar).

Disciplina: Diseño.

Atributos relacionados: Corrección, Reusabilidad.

Referencia: Riel [82]

HEURÍSTICA 049

Nombre: Heurística 3.7

Descripción:

Elimine clases irrelevantes de su sistema.

Disciplina: Diseño.

Atributos relacionados: Corrección, Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 050

Nombre: Heurística 3.8

Descripción:

Elimine clases que están fuera de su sistema.

Disciplina: Diseño.

Atributos relacionados: Corrección, Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 051

Nombre: Heurística 3.9

Descripción:

No cambie una operación por una clase. Sospeche de cualquier clase cuyo nombre es un verbo o es derivado de un verbo, especialmente aquellos que tienen sólo una pieza de comportamiento significativo (i.e., sin contar sets, gets, and prints). Pregunte si esa pieza de comportamiento significativo necesita ser migrada a alguna clase existente o no descubierta.

Disciplina: Diseño, Implementación.

Atributos relacionados: Mantenibilidad, Reusabilidad.

Referencia: Riel [82]

HEURÍSTICA 052

Nombre: Heurística 3.10

Descripción:

Las clases agente son a menudo puestas en el modelo de análisis de una aplicación. Durante el tiempo de diseño, muchos agentes encuentran ser no relevantes y deberían ser removidos.

Disciplina: Diseño.

Atributos relacionados: Corrección.

Referencia: Riel [82]

HEURÍSTICA 053

Nombre: Heurística 4.1

Descripción:

Minimice el número de clases con las cuales otra clase colabora.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad.

Referencia: Riel [82]

HEURÍSTICA 054

Nombre: Heurística 4.2

Descripción:

Minimice el número de mensajes enviados entre una clase y sus colaboradores.

Disciplina: Diseño.

Atributos relacionados: Eficiencia, Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 055

Nombre: Heurística 4.3

Descripción:

Minimice la cantidad de colaboraciones entre una clase y sus colaboradores, esto es, el número de mensajes diferentes enviados.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Corrección.

Referencia: Riel [82]

HEURÍSTICA 056

Nombre: Heurística 4.4

Descripción:

Minimice el abanico en una clase, esto es, el producto del número de mensajes definidos por una clase y los mensajes que envía.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Corrección.

Referencia: Riel [82]

HEURÍSTICA 057

Nombre: Heurística 4.5

Descripción:

Si una clase contiene objetos de otra clase, luego la clase contenedora debería estar enviando mensajes a los objetos contenidos, esto es, la relación de contenido debería implicar siempre una relación de uso.

Disciplina: Diseño.

Atributos relacionados: Interoperabilidad, Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 058

Nombre: Heurística 4.6

Descripción:

La mayoría de los métodos definidos sobre una clase deberían usarse la mayoría de los datos miembro la mayoría del tiempo.

Disciplina: Diseño.

Atributos relacionados: Corrección.

Referencia: Riel [82]

HEURÍSTICA 059

Nombre: Heurística 4.7

Descripción:

Las clases no deberían contener más objetos que un desarrollador pueda adaptar en su memoria de corto plazo. Un valor favorito para éste es el número 6.

Disciplina: Diseño, Implementación.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 060

Nombre: Heurística 4.8

Descripción:

Distribuya la inteligencia del sistema verticalmente hacia abajo resaltando y profundizando las jerarquías que contiene.

Disciplina: Diseño, Implementación.

Atributos relacionados: Mantenibilidad.

Referencia: Riel [82]

HEURÍSTICA 061

Nombre: Heurística 4.9

Descripción:

Cuando implemente restricciones semánticas, es mejor implementarlas en términos de la definición de la clase. A menudo esto encabeza una proliferación de clases, en cuyo caso las restricciones deberían ser implementadas en el comportamiento de la clase-usualmente, pero no necesariamente, en el constructor.

Disciplina: Diseño, Implementación.

Atributos relacionados: Mantenibilidad, Corrección.

Referencia: Riel [82]

HEURÍSTICA 062

Nombre: Heurística 4.10

Descripción:

Cuando implemente restricciones semánticas en el constructor de una clase, coloque restricciones de prueba en el constructor en la jerarquía descendiente de restricciones como el dominio lo permita.

Disciplina: Diseño, Implementación.

Atributos relacionados: Corrección, Interoperabilidad.

Referencia: Riel [82]

HEURÍSTICA 063

Nombre: Heurística 4.11

Descripción:

La información semántica sobre la cual una restricción está basada es mejor ponerla en un objeto central, los objetos de terceros tienen esa información volátil.

Disciplina: Diseño, Implementación.

Atributos relacionados: Corrección, Interoperabilidad.

Referencia: Riel [82]

HEURÍSTICA 064

Nombre: Heurística 4.12

Descripción:

La información semántica sobre la cual una restricción está basada es mejor descentralizarla entre las clases involucradas en la restricción cuando la información es estable.

Disciplina: Diseño, Implementación.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 065

Nombre: Heurística 4.13

Descripción:

Una clase debe conocer qué contiene, pero nunca debería saber quién la contiene.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 066

Nombre: Heurística 4.14

Descripción:

Los objetos que comparten alcance léxico-aquellos contenidos en la misma clase contenedora- no deberían haber usado relaciones entre ellos.

Disciplina: Diseño.

Atributos relacionados: Reusabilidad.

Referencia: Riel [82]

HEURÍSTICA 067

Nombre: Heurística 5.1

Descripción:

La Herencia debería ser usada solamente para modelar una jerarquía de especialización.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 068

Nombre: Heurística 5.2

Descripción:

Las clases derivadas deben tener conocimiento de sus clases base por definición, pero las clases base deberían no saber cualquier cosa sobre sus clases derivadas.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 069

Nombre: Heurística 5.3

Descripción:

Todos los datos en una clase base deberían ser privados; no use datos protegidos.

Disciplina: Diseño, Implementación.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 070

Nombre: Heurística 5.4

Descripción:

En teoría, las jerarquías de Herencia deberían ser profundas- entre más profundo, mejor.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Corrección.

Referencia: Riel [82]

HEURÍSTICA 071

Nombre: Heurística 5.5

Descripción:

En la práctica, las jerarquías de Herencia deberían ser no tan profundas que una persona promedio pueda conservar en su memoria de corto tiempo. Un valor común para esta profundidad es seis.

Disciplina: Diseño, Implementación.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 072

Nombre: Heurística 5.6

Descripción:

Todas las clases abstractas deben ser clases base.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 073

Nombre: Heurística 5.7

Descripción:

Todas las clases base deberían ser clases abstractas.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad.

Referencia: Riel [82]

HEURÍSTICA 074

Nombre: Heurística 5.8

Descripción:

Considere la similitud de datos, comportamiento y/o interfaces tan grande como sea posible en la jerarquía de Herencia.

Disciplina: Diseño.

Atributos relacionados: Corrección, Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 075

Nombre: Heurística 5.9

Descripción:

Si dos o más clases comparten solamente datos comunes (no comportamiento común), entonces esos datos comunes deberían ser puestos en una clase que estará contenida por cada una de las clases compartidas.

Disciplina: Diseño.

Atributos relacionados: Corrección, Interoperabilidad, Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 076

Nombre: Heurística 5.10

Descripción:

Si dos o más clases tienen datos y comportamiento común (i.e., métodos), entonces aquellas clases deberían heredar de una clase base común que capte aquellos datos y comportamiento.

Disciplina: Diseño.

Atributos relacionados: Corrección, Interoperabilidad, Mantenibilidad, Reusabilidad, Testeabilidad

Referencia: Riel [82]

HEURÍSTICA 077

Nombre: Heurística 5.11

Descripción:

Si dos o más clases comparten solamente una interfaz común (i.e., mensajes, no métodos), entonces deberían heredar de una clase base común solamente si serán usadas polimórficamente.

Disciplina: Diseño.

Atributos relacionados: Corrección, Interoperabilidad, Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 078

Nombre: Heurística 5.12

Descripción:

El análisis de casos explícito sobre un tipo de un objeto es usualmente un error. El diseñador debería usar Polimorfismo en la mayoría de los casos.

Disciplina: Análisis, Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 079

Nombre: Heurística 5.13

Descripción:

El análisis de casos explícito sobre un valor de un atributo es a menudo un error. La clase debería estar descompuesta en una jerarquía de Herencia, donde cada valor de un atributo es transformado en una clase derivada.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 080

Nombre: Heurística 5.14

Descripción:

No modele la semántica dinámica de una clase a través del uso de una relación de Herencia. Un intento para modelar la semántica dinámica con una relación semántica estática le guiará a una alternancia de tipos en tiempo de ejecución.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 081

Nombre: Heurística 5.15

Descripción:

NO cambie objetos de una clase hacia clases derivadas de la clase. Sospeche de cualquier clase derivada para la cual solo hay una instancia.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad.

Referencia: Riel [82]

HEURÍSTICA 082

Nombre: Heurística 5.16

Descripción:

Si piensa que necesita crear nuevas clases en tiempo de ejecución, tome un paso atrás para darse cuenta que lo que está tratando de crear son objetos. Ahora generalice aquellos objetos en una clase.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 083

Nombre: Heurística 5.17

Descripción:

Sería ilegal para una clase derivada sobrescribir un método de una clase base con un método NOP, esto es, un método que no hace nada.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad.

Referencia: Riel [82]

HEURÍSTICA 084

Nombre: Heurística 5.18

Descripción:

NO confunda contención opcional con la necesidad por Herencia. El modelado de contención opcional con Herencia le guiará hacia una proliferación de clases.

Disciplina: Diseño.

Atributos relacionados: Reusabilidad.

Referencia: Riel [82]

HEURÍSTICA 085

Nombre: Heurística 5.19

Descripción:

Cuando construya una jerarquía de Herencia, trate de construir frameworks reutilizables en lugar de componentes reutilizables.

Disciplina: Diseño, Implementación.

Atributos relacionados: Interoperabilidad, Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 086

Nombre: Heurística 6.1

Descripción:

Si tiene un ejemplo de Herencia múltiple en su diseño, asuma que tiene un error y luego pruebe otra forma.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 087

Nombre: Heurística 6.2

Descripción:

Cada vez que haya Herencia en un diseño orientado a objetos, hágase usted mismo dos cuestionamientos:

1. ¿Soy un tipo especial de la cosa desde la cual estoy heredando?
2. ¿Es la cosa desde la que estoy heredando parte de mí?

Disciplina: Diseño.

Atributos relacionados: Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 088

Nombre: Heurística 6.3

Descripción:

Cada vez que ha encontrado una relación de Herencia múltiple en un diseño orientado a objetos, asegúrese que ninguna clase base es actualmente una clase derivada de otra clase base.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 089

Nombre: Heurística 7.1

Descripción:

Cuando dada una alternativa en un diseño orientado a objetos entre una relación de contención y una relación de asociación, elija la relación de contención.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 090

Nombre: Heurística 8.1

Descripción:

No use datos globales o funciones para desarrollar contabilidad de información sobre objetos de una clase. Debería usar métodos o variables de clase en su lugar.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 091

Nombre: Heurística 9.1

Descripción:

Los diseñadores orientados a objetos deberían no permitir criterios de diseño físico que corrompan sus diseños lógicos. Sin embargo, los criterios de diseño físico a menudo son usados en el proceso de elaboración de decisiones en tiempo de diseño lógico.

Disciplina: Diseño.

Atributos relacionados: Desempeño, Corrección, Interoperabilidad, Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 092

Nombre: Heurística 9.2

Descripción:

No cambie el estado de un objeto sin ir a través de su interfaz pública.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Riel [82]

HEURÍSTICA 093

Nombre: Heurística1

Descripción:

Observa las tendencias en las características de las clases y dependencias de las clases de su sistema antes de intentar cualquier refactorización.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Garzas [57]

HEURÍSTICA 094

Nombre: Heurística2

Descripción:

Acepte que la automatización es realista para un conjunto relativamente pequeño de refactorizaciones. Muchas de las más complejas refactorizaciones pueden ser alcanzadas sólo con herramientas de soporte.

Disciplina: Implementación.

Atributos relacionados: Mantenibilidad, Reutilización.

Referencia: Garzas [57]

HEURÍSTICA 095

Nombre: Heurística3

Descripción:

Al interior de una jerarquía de Herencia, las dependencias de la clase descendiente deberían ser una consideración principal en la elaboración de cualquier decisión de refactorización.

Disciplina: Mantenimiento.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: Garzas [57]

HEURÍSTICA 096

Nombre: Heurística4

Descripción:

El acoplamiento (en lugar de cohesión) debería ser la característica de una clase que intenta optimizarse.

Disciplina: Mantenimiento.

Atributos relacionados: Mantenibilidad, Reusabilidad, Confiabilidad, Desempeño.

Referencia: Garzas [57]

HEURÍSTICA 097

Nombre: Heurística5

Descripción:

Un esfuerzo consistente debería ser aplicado al proceso de refactorización cada vez que sea posible; el crecimiento del sistema también debería ser monitoreado.

Disciplina: Mantenimiento.

Atributos relacionados:

Referencia: Garzas [57]

HEURÍSTICA 098

Nombre: Heurística6

Descripción:

Cuando considere refactorización alguna, necesitamos apreciar que otras refactorizaciones quizás sean necesarias dado una dependencia entre refactorizaciones.

Disciplina: Mantenimiento.

Atributos relacionados:

Referencia: Garzas [57]

HEURÍSTICA 099

Nombre: Operabilidad

Descripción:

Lo mejor es que trabaja, lo más eficientemente es que pueda ser probado. El sistema debería tener pocos errores, ningún error debería bloquear la ejecución de pruebas y el producto debería evolucionar en estados funcionales (en simultánea al desarrollo y pruebas).

Disciplina: Pruebas.

Atributos relacionados: Confiabilidad, Desempeño, Testeabilidad.

Referencia: AJITHA [99]

HEURÍSTICA 100

Nombre: Capacidad de observación

Descripción:

Lo que vemos es lo que probamos.

- Distintas salidas deberían ser generadas para cada entrada.
- El estado del sistema actual y pasado y las variables deberían estar visibles durante las pruebas
- Todos los factores que afectan las salidas deberían ser visibles
- Las salidas incorrectas deberían ser fácilmente identificadas.
- El código fuente debería ser fácilmente accesible.

Los errores internos deberían ser automáticamente detectados (a través de mecanismos de auto-prueba) y reportados.

Disciplina: Pruebas.

Atributos relacionados: Confiabilidad, Desempeño, Funcionalidad, Usabilidad, Corrección, Testeabilidad

Referencia: AJITHA [99]

HEURÍSTICA 101

Nombre: Capacidad de control

Descripción:

Lo mejor que controlamos es el software, por demás el proceso de pruebas puede ser automatizado y optimizado.

Verifique que:

- Todas las salidas pueden ser generadas y el código puede ser ejecutado a través de una combinación de entradas.
- Los estados de software y hardware pueden ser controlados directamente por el ingeniero de pruebas.
- Los formatos de entradas y salidas son consistentes y Estructurados

La prueba puede ser convenientemente especificada, automatizada y reproducida

Disciplina: Pruebas.

Atributos relacionados: Confiabilidad, Desempeño, Testeabilidad.

Referencia: AJITHA [99]

HEURÍSTICA 102

Nombre: Capacidad de descomposición

Descripción:

A través del control de pruebas, podemos rápidamente aislar problemas y desarrollar pruebas eficientes y efectivas.

El sistema software debería ser construido desde módulos independientes que puedan ser probados independientemente.

Disciplina: Pruebas.

Atributos relacionados: Confiabilidad, Desempeño, Testeabilidad.

Referencia: AJITHA [99]

HEURÍSTICA 103

Nombre: Simplicidad

Descripción:

Lo menos aquí es probar, lo más es probarlo rápidamente.

Los puntos a considerar en este respecto son simplicidad funcional (e.g. conjunto mínimo de características), estructural (e.g., la arquitectura está modularizada) y de código (e.g., un estándar de codificación es adoptado).

Disciplina: Pruebas.

Atributos relacionados: Confiabilidad, Desempeño, Testeabilidad.

Referencia: AJITHA [99]

HEURÍSTICA 104

Nombre: Estabilidad

Descripción:

Siendo pocos los cambios, pocos son los trastornos para probar.

Los cambios en el software deberían ser no frecuentes, controlados y no invalidando pruebas existentes. El software debería ser capaz de recuperarse bien ante fallas.

Disciplina: Pruebas.

Atributos relacionados: Confiabilidad, Desempeño, Testeabilidad.

Referencia: AJITHA [99]

HEURÍSTICA 105

Nombre: Comprensibilidad

Descripción:

La mayor información que tendremos, la más inteligente que probaremos.

Los probadores deberían ser capaces de entender bien el diseño, los cambios en el diseño y las dependencias entre componentes internos, externos y compartidos.

La documentación técnica debería ser accesible al instante, precisa, bien organizada, específica y detallada.

Disciplina: Pruebas.

Atributos relacionados: Funcionalidad, Usabilidad, Corrección, Interoperabilidad, Testeabilidad.

Referencia: AJITHA [99]

HEURÍSTICA 106

Nombre: Apropiabilidad

Descripción:

Lo mayor que sabemos sobre la intención de uso del software, es que lo mejor que podemos organizar es nuestra prueba para encontrar errores.

Disciplina: Pruebas.

Atributos relacionados: Mantenibilidad, Testeabilidad.

Referencia: AJITHA [99]

HEURÍSTICA 107

Nombre: Realimentación

Descripción:

El sistema debe siempre mantener a los usuarios informados del estado del sistema, con una realimentación apropiada y en un tiempo razonable.

Disciplina: Pruebas, Mantenimiento.

Atributos relacionados: Confiabilidad, Usabilidad, Testeabilidad.

Referencia: FERNÁNDEZ [7]

HEURÍSTICA 108

Nombre: Gestionando el error

Descripción:

El sistema debe prevenir la aparición de errores antes que generar buenos mensajes de alerta; hay dos formas de gestionar el error: una es previniendo que el error ocurra y la otra es corrigiendo el error. En caso de que el usuario cometa un error el sistema debe proveer mensajes expresados en un lenguaje claro (sin códigos extraños), indicando exactamente el problema de manera constructiva.

Disciplina: Pruebas, Mantenimiento.

Atributos relacionados: Mantenibilidad, Testeabilidad.

Referencia: FERNÁNDEZ [7]

HEURÍSTICA 109

Nombre: Consistencia

Descripción:

El sistema debe prevenir que los usuarios tengan que preguntarse si las diversas palabras, situaciones, o acciones significan la misma cosa. La consistencia puede darse de las siguientes maneras:

a. Visual: Los elementos de la interfaz de usuario deben ser consecuentes o consistentes en el aspecto y estructura.

b. Funcional: La manera de llevar a cabo las diferentes tareas a través del sistema debe ser consecuente, con otros sistemas similares, e incluso entre diferente clases de aplicaciones en el mismo sistema.

c. Evolutiva: Debe ser fácil tener una nueva versión del software, y que esta sea consistente con las anteriores. Como por ejemplo, en el caso de una familia de producto software, la consistencia sobre los productos en la familia es un aspecto importante.

Disciplina: Pruebas, Mantenimiento.

Atributos relacionados: Confiabilidad, Funcionalidad, Usabilidad, Testeabilidad.

Referencia: FERNÁNDEZ [7]

HEURÍSTICA 110

Nombre: Ayuda u orientación

Descripción:

Aunque es mejor si el sistema puede usarse sin documentación, puede ser necesario disponer de ayuda y documentación. Ésta debe ser fácil de buscar, centrada en las tareas del usuario, tener información de las etapas a realizar, ser contextual y no muy extensa.

Disciplina: Pruebas, Mantenimiento.

Atributos relacionados: Usabilidad.

Referencia: FERNÁNDEZ [7].

HEURÍSTICA 111

Nombre: Minimizar la carga cognitiva.

Descripción:

Los seres humanos tienen limitaciones cognitivas del tratamiento de la información en memoria a corto plazo, por ello requiere que lo mostrado por pantalla sea simple, por ejemplo presentar siete ítems por pantalla es una sobrecarga de información porque según estudios realizados siete u ocho es el medio límite de memoria a corto plazo. De igual forma el usuario no debe tener que recordar la información de una parte de diálogo a la otra; es mejor mantener objetos, acciones, y opciones visibles que memorizar.

Disciplina: Pruebas, Mantenimiento.

Atributos relacionados: Confiabilidad, Usabilidad, Testeabilidad.

Referencia: FERNÁNDEZ [7]

HEURÍSTICA 112

Nombre: Explícito control de usuario

Descripción:

Los usuarios desean tener el control total del sistema y que el sistema responda a sus acciones. El diseño del sistema debe responder a las acciones de los usuarios, y que estos sean los iniciadores de las acciones, no solo los que respondan a acciones del sistema. El usuario debe tener la impresión de que él tiene "el control" de la aplicación.

Disciplina: Pruebas, Mantenimiento.

Atributos relacionados: Usabilidad.

Referencia: FERNÁNDEZ [7]

HEURÍSTICA 113

Nombre: Mapeo natural

Descripción:

El sistema debe proveer una relación clara entre lo que el usuario quiere hacer y el mecanismo para hacerlo. Esta propiedad puede estar estructurada de la siguiente manera:

- a. Predecible: Para el usuario el comportamiento del sistema debe ser predecible.
- b. Lenguaje de los usuarios: El sistema debe hablar el lenguaje de los usuarios, con las palabras, las frases y los conceptos familiares, en lugar de que los términos estén orientados al sistema. Utilizar convenciones del mundo real, haciendo que la información aparezca en un orden natural y lógico.
- c. Facilidad de la navegación: Debe ser obvio al usuario cómo navegar en el sistema.

Disciplina: Pruebas, Mantenimiento.

Atributos relacionados: Confiabilidad, Funcionalidad, Usabilidad, Testeabilidad.

Referencia: FERNÁNDEZ [7]

HEURÍSTICA 114

Nombre: Accesibilidad

Descripción:

Los sistemas deben ser accesibles en todas las maneras que es requerido. Tal propiedad podría ser descompuesta de la siguiente manera:

- a. Discapacidades: los sistemas deben suministrar el soporte para usuarios que son discapacitados (/ ciego / sordo/ etc.).
- b. Multicanal: el sistema debe ser capaz de soportar el acceso por varios medios de comunicación. Este es un concepto muy amplio que varía desde ser capaz de navegar en un sitio Web por medio de un teléfono o ser capaz de hacerlo a través de audio.

Disciplina: Pruebas.

Atributos relacionados: Usabilidad.

Referencia: FERNÁNDEZ [7]

HEURÍSTICA 115

Nombre: Heurística de dinamicidad

Descripción:

Un modelo completo tendría necesariamente un diagrama de estado para cada clase. Esta suposición no siempre cumple por dos razones: no es práctico crear diagramas de estado para cada clase y el comportamiento interno no está al mismo nivel dinámico alto para todas las clases. El criterio para decidir si una clase necesita un diagrama de estado podría por eso ser dependiente de su nivel dinámico interno – su dinamicidad. Definimos la dinamicidad como la cantidad de mensajes entrantes y salientes de las instancias de la clase a lo largo de Flujo de secuencia de mensajes.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Corrección.

Referencia: LANGE [100]

HEURÍSTICA 116

Nombre: Usando el modelo de proceso

Descripción:

Para que la Reutilización sea convertida en una parte implícita de la ingeniería de proceso, un número significativo o masivo crítico de componentes debe existir en un espacio de desarrollo dado. En otras palabras, un número variable de productos reutilizables debe existir antes de que los beneficios técnicos y económicos de la Reutilización sean tangibles.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad.

Referencia: CORNWELL [45]

HEURÍSTICA 117

Nombre: Herencia de hilos

Descripción:

La Herencia es una técnica de modificación controlada. Sin embargo, está abierta al abuso.

El uso indisciplinado de incrementos y redefiniciones puede alterar significativamente las propiedades funcionales y no funcionales de una clase heredada. Tanto así que, las últimas generaciones de la subclase contendrán similitudes pequeñas o no a la superclase original.

La superclase base debería ser usada para establecer una Abstracción principal. Una especificación del Desempeño y comportamiento que esta preservado en todas las generaciones subsecuentes de subclases y no está sujeto a

modificaciones. Una subclase representaría una especialización incremental de una Abstracción principal.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Corrección.

Referencia: CORNWELL [45]

HEURÍSTICA 118

Nombre: Diseño de clase abstracta

Descripción:

Una clase abstracta es una interfaz reutilizable. Representa un conjunto de obligaciones de diseño funcionales y no funcionales que deben satisfacerse por un objeto concreto, poblando una instancia de esa clase. Una clase abstracta es típicamente un producto de un dominio de ingeniería. Está construido cuando un análisis de dominio identifica la necesidad por un componente que represente una interfaz común y procesamiento de Abstracción. Pero es probable que requiera una implementación separada para contexto puesto ahí.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad.

Referencia: CORNWELL [45]

HEURÍSTICA 119

Nombre: Corrección

Descripción:

Un componente solo puede ser juzgado como completamente correcto, cuando es puesto en un contexto y sometido a pruebas de tiempo y cronograma en el espacio de análisis. Sin embargo, es posible verificar que un componente es correcto con respecto a su propia implementación.

Disciplina: Pruebas, Mantenimiento.

Atributos relacionados: Corrección, Interoperabilidad, Testeabilidad.

Referencia: CORNWELL [45]

HEURÍSTICA 120

Nombre: Heurística 1

Descripción:

Las propiedades de la clase deberían estar ocultas detrás de métodos de acceso. Esto significa que las propiedades de la clase no deberían ser usadas en la interfaz pública de una clase. La interfaz pública de una clase es una lista de métodos y atributos que podrían ser utilizados desde el exterior de la clase. El Mantenimiento es el principal motivo de estas heurísticas. El esfuerzo consistente de ocultamiento de información en los niveles de diseño e implementación es responsable de gran parte de los beneficios del paradigma orientado a objetos. Estas heurísticas son usadas en las fases de análisis, diseño y codificación. La meta principal de estas heurísticas es el Mantenimiento y éstas heurísticas ayudan a los diseñadores para modificar el comportamiento e implementaciones más libremente.

Disciplina: Análisis.

Atributos relacionados: Flexibilidad, Eficiencia, Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: BERNÁRDEZ [81]

HEURÍSTICA 121

Nombre: Heurística 2

Descripción:

El principal motivo de estas heurísticas es intentar conservar el número de funciones miembro a un número manejable tan bien como limitar el número de atributos al interior de una clase. “Conservando la interfaz mínima, el sistema será fácil de entender y los componentes fáciles de reutilizar” (Heurística #2.4 en Riel [82]). Habrá muchas responsabilidades para aquellas clases que tienen muchos métodos y por esta razón el sistema será más complejo y difícil de mantener.

Disciplina: Diseño.

Atributos relacionados: Flexibilidad, Eficiencia, Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: BASHAR [48]

HEURÍSTICA 122

Nombre: Heurística 3

Descripción:

Una clase capturaría una y solo una Abstracción clave con toda su información y todo su comportamiento. Una entidad principal al interior de un modelo de dominio es conocido como una Abstracción clave [1]. “Si el modelo tiene más de una Abstracción debería involucrar más de una clase. Algunas veces es difícil determinar todas las clases afectadas. A menudo algunas de esas clases son muy pequeñas y pueden ser eliminadas”. (Heurística #5 en el apéndice B). Riel plantea que hay una posibilidad que el diseñador ha capturado que cada función es a una clase. Combinando métodos y atributos en una única clase debería resultar en una clase muy grande, aún cuando sería un modelo de Abstracción correcto.

Disciplina: Diseño.

Atributos relacionados: Eficiencia, Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: BASHAR [48]

HEURÍSTICA 123

Nombre: Heurística 4

Descripción:

Estas heurísticas mencionan, que teniendo muchos mensajes entre clases haciendo mucho o métodos que tienen mala granularidad y por esta razón el sistema será complejo de mantener. Con grandes interfaces públicas, también para los diseñadores es difícil mantener encontrar un método requerido fácilmente, así hace la clase difícil de entender y más compleja. Una clase que captura más de un conjunto de abstracciones sugiere que debería ser una clase eliminada. En este aspecto Riel menciona que la mayoría de métodos de clase utiliza la mayoría de atributos de clase.

Disciplina: Diseño.

Atributos relacionados: Interoperabilidad, Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: BASHAR [48]

HEURÍSTICA 124

Nombre: Heurística 5

Descripción:

Estas heurísticas mencionan que una vez el número de colaboraciones es controlado, luego hace tener sentido para minimizar la cantidad de colaboración entre clases.

Si esas heurísticas consideradas igual a la heurística de minimización de colaboración, entonces significa que hay un punto de corte entre adicionar más mensajes a enviar entre clases y adicionar una nueva colaboración. El monto de acoplamiento es un pequeño problema compara con la Complejidad de una nueva colaboración.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad.

Referencia: BASHAR [48]

HEURÍSTICA 125

Nombre: Heurística 6

Descripción:

Estas heurísticas sugieren que propiedades comunes podrían ser una Abstracción única, una clase y una instancia. Esto hace la propiedad más centralizada y reutilizable. Dittrich R hace mención que su las propiedades son comunes definir las en diferentes locaciones resulta en redundancias no deseadas. Estas redundancias encabezan un esfuerzo innecesario si las propiedades tienen que ser cambiadas. En este sentido, Riel plantea un ejemplo que si todas las válvulas se activaran y desactivaran, pero cada válvula lo hace de manera diferente, entonces la clase base "válvula" no es útil a menos que exista algún lugar en la aplicación donde una válvula genérica necesitara decidir para lo cual esto es en tiempo de ejecución.

Disciplina: Diseño, Implementación.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: BASHAR [48]

HEURÍSTICA 126

Nombre: Heurística 7

Descripción:

Estas heurísticas discuten, que todas las raíces de un árbol de Herencia deberían ser abstractas. Desde la clase raíz es posiblemente la clase más importante como que provee una interfaz. Las clases bases abstractas que distribuyen la implementación se convierten en problemáticas en tanto el tamaño de la jerarquía de Herencia se incrementa. Entonces la raíz de la jerarquía de Herencia debería ser una definición de tipo abstracto.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: BASHAR [48]

HEURÍSTICA 127

Nombre: Heurística B1

Descripción:

Una clase en una jerarquía de contenido solamente dependería de sus clases hijas.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: BASHAR [48]

HEURÍSTICA 128

Nombre: Heurística B2

Descripción:

Cada atributo debería estar oculto con su clase.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: BASHAR [48]

HEURÍSTICA 129

Nombre: Heurística B3

Descripción:

Una dependencia cliente-servidor entre dos clases debería no encabezar dependencias desde el servidor al cliente.

Disciplina: Diseño.

Atributos relacionados: Reusabilidad, Corrección, Interoperabilidad.

Referencia: BASHAR [48]

HEURÍSTICA 130

Nombre: Heurística B4

Descripción:

Elimine dependencias desde clases de base de datos a sus clientes.

Disciplina: Análisis, Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: BASHAR [48]

HEURÍSTICA 131

Nombre: Heurística B5

Descripción:

Una clase debería capturar una y solamente una Abstracción clave con toda su información y todo su comportamiento.

Disciplina: Diseño.

Atributos relacionados: Corrección, Eficiencia, Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: BASHAR [48]

HEURÍSTICA 132

Nombre: Heurística B6

Descripción:

No cree clases innecesarias para modelar roles.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: BASHAR [48]

HEURÍSTICA 133

Nombre: Heurística B7

Descripción:

Elimine métodos de acceso puros.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad.

Referencia: BASHAR [48]

HEURÍSTICA 134

Nombre: Heurística B8

Descripción:

Elimine relaciones adicionales desde clases base hacia sus clases derivadas.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: BASHAR [48]

HEURÍSTICA 135

Nombre: Heurística B9

Descripción:

Elimine clases con propiedades que implican redundancias.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: BASHAR [48]

HEURÍSTICA 136

Nombre: Heurística B10

Descripción:

Elimine dependencias multivaluadas.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Corrección.

Referencia: BASHAR [48]

HEURÍSTICA 137

Nombre: Heurística B11

Descripción:

Convierta asociaciones y relaciones de uso en la relación de contención más fuerte dondequiera que sea posible.

Disciplina: Diseño, Implementación.

Atributos relacionados: Reusabilidad, Corrección.

Referencia: BASHAR [48]

HEURÍSTICA 138

Nombre: Heurística B12

Descripción:

Elimine las instancias contenidas que tienen que ser modificadas constantemente.

Disciplina: Implementación.

Atributos relacionados: Mantenibilidad, Corrección.

Referencia: BASHAR [48]

HEURÍSTICA 139

Nombre: Heurística B13

Descripción:

Todas las propiedades de la interfaz clase base deben ser usables en instancias de sus clases derivadas en cualquier punto donde una clase base es esperada.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: BASHAR [48]

HEURÍSTICA 140

Nombre: Heurística B14

Descripción:

Propiedades comunes de instancias deberían estar definidas en una localización única.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: BASHAR [48]

HEURÍSTICA 141

Nombre: Heurística B15

Descripción:

Clases inestables no deberían ser clases base.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Corrección.

Referencia: BASHAR [48]

HEURÍSTICA 142

Nombre: Heurística B16

Descripción:

No haga uso indebido de Herencia para compartir atributos.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Corrección.

Referencia: BASHAR [48]

HEURÍSTICA 143

Nombre: Heurística B17

Descripción:

La sobrecarga debería definir solamente diferencias para métodos sobrecargados.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Corrección.

Referencia: BASHAR [48]

HEURÍSTICA 144

Nombre: Heurística B18

Descripción:

Elimine análisis de caso sobre propiedades de instancias.

Disciplina: Diseño.
Atributos relacionados: Mantenibilidad, Corrección.
Referencia: BASHAR [48]

HEURÍSTICA 145

Nombre: Heurística B19
Descripción:
Prefiera escribir por atributo antes de escribir por Herencia.
Disciplina: Diseño.
Atributos relacionados: Mantenibilidad, Corrección.
Referencia: BASHAR [48]

HEURÍSTICA 146

Nombre: Heurística B20
Descripción:
Un método debería usar solamente clases de atributos de su clase, clases de sus parámetros o clases de instancias creadas localmente.
Disciplina: Diseño.
Atributos relacionados: Mantenibilidad, Corrección.
Referencia: BASHAR [48]

HEURÍSTICA 147

Nombre: Heurística CA1
Descripción:
En la agregación, el agregado debería limitar el número de agregados.
Disciplina: Diseño, Implementación.
Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.
Referencia: BASHAR [48]

HEURÍSTICA 148

Nombre: Heurística CA2
Descripción:
Limite el número de agregados definidos por el usuario.
Disciplina: Diseño.
Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.
Referencia: BASHAR [48]

HEURÍSTICA 149

Nombre: Heurística CA3
Descripción:
Restrinja el acceso directo a agregados por el usuario.
Disciplina: Diseño.
Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.
Referencia: BASHAR [48]

HEURÍSTICA 150

Nombre: Heurística CA4
Descripción:
Limite los mecanismos de habilitación que violan la encapsulación.

Disciplina: Diseño.
Atributos relacionados: Mantenibilidad, Corrección.
Referencia: BASHAR [48]

HEURÍSTICA 151

Nombre: Heurística CA5
Descripción:
La agregación de nodos hojas son pequeños, reutilizables y simples.
Disciplina: Diseño.
Atributos relacionados: Mantenibilidad, Corrección.
Referencia: BASHAR [48]

HEURÍSTICA 152

Nombre: Heurística CA6
Descripción:
Cree jerarquías de agregación que son estrechas y profundas.
Disciplina: Diseño.
Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.
Referencia: BASHAR [48]

HEURÍSTICA 153

Nombre: Heurística CA7
Descripción:
Las jerarquías de agregación deberían no ser muy profundas.
Disciplina: Diseño.
Atributos relacionados: Mantenibilidad, Corrección.
Referencia: BASHAR [48]

HEURÍSTICA 154

Nombre: Heurística CA8
Descripción:
La Estabilidad debería descender entre la jerarquía
Disciplina: Diseño.
Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.
Referencia: BASHAR [48]

HEURÍSTICA 155

Nombre: Heurística CI1
Descripción:
Limite el uso de Herencia múltiple.
Disciplina: Diseño.
Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.
Referencia: BASHAR [48]

HEURÍSTICA 156

Nombre: Heurística CI2
Descripción:
La jerarquía de Herencia debería no ser más de seis en profundidad.
Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: BASHAR [48]

HEURÍSTICA 157

Nombre: Heurística CI3

Descripción:

La raíz de la jerarquía de Herencia debería ser abstracta.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: BASHAR [48]

HEURÍSTICA 158

Nombre: Heurística CI4

Descripción:

La jerarquía de Herencia es una jerarquía de especialización.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad.

Referencia: BASHAR [48]

HEURÍSTICA 159

Nombre: Heurística CI5

Descripción:

Minimice los cortes en las jerarquías tipo/clase.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Corrección.

Referencia: BASHAR [48]

HEURÍSTICA 160

Nombre: Heurística CI6

Descripción:

Las clases abstractas raíz deberían ser definiciones de tipo.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Corrección, Reusabilidad, Testeabilidad.

Referencia: BASHAR [48]

HEURÍSTICA 161

Nombre: Heurística CI7

Descripción:

Prevenga la sobre-generalización de una superclase.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Corrección.

Referencia: BASHAR [48]

HEURÍSTICA 162

Nombre: Heurística CI8

Descripción:

Haga abstracciones tantos muchos nodos intermediarios como sea posible.

Disciplina: Análisis, Diseño.

Atributos relacionados: Mantenibilidad, Corrección, Reusabilidad.

Referencia: BASHAR [48]

HEURÍSTICA 163

Nombre: Heurística CI9

Descripción:

La Estabilidad debería ascender en la jerarquía.

Disciplina: Análisis, Diseño.

Atributos relacionados: Mantenibilidad, Corrección, Reusabilidad.

Referencia: BASHAR [48]

HEURÍSTICA 164

Nombre: Heurística CU1

Descripción:

Limite el número de colaboraciones entre clases.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: BASHAR [48]

HEURÍSTICA 165

Nombre: Heurística CU2

Descripción:

Minimice la visibilidad de colaboradores de interfaces.

Disciplina: Diseño, Implementación.

Atributos relacionados: Mantenibilidad, Corrección.

Referencia: BASHAR [48]

HEURÍSTICA 166

Nombre: Heurística CU3

Descripción:

Identifique y estabilice colaboradores en lugares comunes.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad.

Referencia: BASHAR [48]

HEURÍSTICA 167

Nombre: Heurística CC1

Descripción:

Limite un ancho de banda absoluto de objetos.

Disciplina: Diseño.

Atributos relacionados:

Referencia: BASHAR [48]

HEURÍSTICA 168

Nombre: Heurística CC2

Descripción:

Limite los atributos internos de un objeto.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Corrección.

Referencia: BASHAR [48]

HEURÍSTICA 169

Nombre: Heurística CC3

Descripción:

Limite una responsabilidad real del objeto.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad.

Referencia: BASHAR [48]

HEURÍSTICA 170

Nombre: Heurística CC4

Descripción:

Minimice métodos complejos.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad.

Referencia: BASHAR [48]

HEURÍSTICA 171

Nombre: Sí hay dependencias de clases concretas

Descripción:

Si hay dependencias de clases concretas, entonces haga que esas dependencias sean de abstracciones, con el propósito de evitar depender de los elementos que puedan ser cambiados.

Disciplina: Diseño, Implementación.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: GARZAS [57]

HEURÍSTICA 172

Nombre: Si un objeto tiene diferente comportamiento según su estado

Descripción:

Si un objeto tiene diferente comportamiento según su estado, entonces colocar cada uno de estos comportamientos en una clase separada, con el propósito de evitar que los servicios se comporten de una u otra forma según el estado de la clase.

Disciplina: Diseño, Implementación.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: GARZAS [57]

HEURÍSTICA 173

Nombre: Si una jerarquía de clases tiene muchos niveles

Descripción:

Si una jerarquía de clases tiene muchos niveles, entonces reducir el uso de la Herencia, con el propósito de evitar jerarquías grandes y difícilmente mantenibles.

Disciplina: Diseño, Mantenimiento.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: GARZAS [57]

HEURÍSTICA 174

Nombre: Si algo se utiliza muy poco o no se utiliza.

Descripción:

Si algo se utiliza muy poco o no se utiliza, entonces elimínelo, con el propósito de evitar que en un diseño existan elementos que no aportan valor o utilidad.

Disciplina: Diseño.

Atributos relacionados: Eficiencia, Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: GARZAS [57]

HEURÍSTICA 175

Nombre: Si una superclase conoce alguna de sus subclases

Descripción:

Si una superclase conoce alguna de sus subclases, entonces elimine ese conocimiento, con el propósito de evitar que la clase padre conozca a alguna de sus clases hijas directas o indirectas.

Disciplina: Diseño.

Atributos relacionados: Eficiencia, Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: GARZAS [57]

HEURÍSTICA 176

Nombre: Si una clase colabora con muchas

Descripción:

Si una clase colabora con muchas, entonces reduzca el número de colaboraciones, con el propósito de evitar que las clases tengan una gran cantidad de colaboración.

Disciplina: Diseño.

Atributos relacionados: Flexibilidad, Eficiencia, Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: GARZAS [57]

HEURÍSTICA 177

Nombre: Si un cambio en una interfaz impacta en muchos clientes.

Descripción:

Si un cambio en una interfaz impacta en muchos clientes, entonces cree interfaces específicas para cada cliente, con el propósito de evitar usar interfaces que soporten a muchos clientes y con distintas necesidades.

Disciplina: Diseño, Mantenimiento.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: GARZAS [57]

HEURÍSTICA 178

Nombre: Si entre una interfaz y su implementación no hay una Abstracción

Descripción:

Si entre una interfaz y su implementación no hay una Abstracción, entonces cree una clase abstracta con una implementación por defecto entre la interfaz y la clase que implementa, con el propósito de evitar el que no existan implementaciones por defecto.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Corrección, Reusabilidad.

Referencia: GARZAS [57]

HEURÍSTICA 179

Nombre: Si una superclase es concreta

Descripción:

Si una superclase es concreta, entonces reestructurarla para eliminarla, con el propósito de evitar que existan superclases concretas.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: GARZAS [57]

HEURÍSTICA 180

Nombre: Si un servicio tiene muchos parámetros

Descripción:

Si un servicio tiene muchos parámetros, entonces crear varios métodos, reduciendo la lista o pasar un objeto, con el propósito de evitarlos, porque son poco comprensibles y aumentan el acoplamiento.

Disciplina: Diseño, Implementación.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: GARZAS [57]

HEURÍSTICA 181

Nombre: Si una clase es grande.

Descripción:

Si una clase es grande, entonces reduzca su tamaño repartiendo Funcionalidad en otras clases, con el propósito de evitar clases grandes y poco cohesionadas.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: GARZAS [57]

HEURÍSTICA 182

Nombre: Si elementos de interfaz de usuario están en entidades de dominio.

Descripción:

Si elementos de interfaz de usuario están en entidades de dominio, entonces coloque estos elementos en una entidad aparte, con el propósito de evitar que las clases de dominio contengan elementos cuyo objetivo es como se representa la información.

Disciplina: Diseño, Implementación.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: GARZAS [57]

HEURÍSTICA 183

Nombre: Si una clase utiliza más cosas de otra que de sí misma.

Descripción:

Si una clase utiliza más cosas de otra que de sí misma, entonces pase esas cosas a la clase que más las utiliza, con el propósito de evitar que la inteligencia de la clase esté distribuida en otras.

Disciplina: Diseño, Implementación.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: GARZAS [57]

HEURÍSTICA 184

Nombre: Si una clase rechaza algo de lo que hereda

Descripción:

Si una clase rechaza algo de lo que hereda, entonces evítelo, generalmente cambiando la Herencia por generalización, con el propósito de evitar que las clases tengan que rechazar lo que heredan de sus superclases.

Disciplina: Diseño, Implementación.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: GARZAS [57]

HEURÍSTICA 185

Nombre: Si los atributos de una clase son públicos o protegidos.

Descripción:

Si los atributos de una clase son públicos o protegidos, entonces hacer estos privados y acceder a ellos a través de servicios, con el propósito de evitar que se pueda acceder a los atributos de un objeto.

Disciplina: Diseño.

Atributos relacionados: Mantenibilidad, Reusabilidad, Testeabilidad.

Referencia: GARZAS [57]

ANEXO J: INFLUENCIAS

Este anexo muestra las relaciones que existen entre los atributos y sub atributos, encontrados y clasificados en claros y grises, según la capa a la que pertenece.

Capa 1	En esta capa estan orgnizados los factores de calidad que son independientes entre si. El atributo es considerado como externo. Representa atributos abstractos.
Capa 2	Esta capa indica todos los factores de calidad cuya valoración contribuye a la valoración de los factores de alto nivel que están en la capa 1; a su vez, estos son atributos externos. Pueden ser compartidos o exclusivos.
Capa 3 C	Se consideran los atributos internos, que tienen influencia sobre los externos. Un atributos <i>Compartido</i> es aquel que le sirve a más de un atributo externo
Capa 3 E	Aquí son considerados los atributos internos, que tienen influencia sobre los externos. Un atributo <i>Exclusivo</i> es aquel que sólo le sirve a un atributo externo o interno únicamente,

Desempeño	Eficiencia
	Consumo de recursos
	Velocidad

.Confiabilidad	Acoplamiento
	Completitud
	Composición
	Conformidad
	Consistencia
	Disponibilidad
	Encapsulamiento
	Exactitud
	Interoperabilidad
	Madurez
	Modularidad
	Recuperabilidad
	Simplicidad
Tolerancia al error	

Flexibilidad	Auto descripción
	Cambiabilidad
	Composición
	Encapsulamiento
	Generalidad
	Modularidad
	Polimorfismo

Funcionalidad	Cohesión
	Conformidad
	Corrección
	Exactitud
	Generalidad
	Idoneidad
	Interoperabilidad
	Jerarquía
	Polimorfismo
	Seguridad
	Tamaño

Portabilidad	Adaptabilidad
	Auto-Descripción
	Cohesión
	Compleitud
	Conformidad
	Consistencia
	Documentación
	Encapsulamiento
	Independencia de Hardware
	Independencia de Software
	Instalabilidad
	Modularidad
	Remplazabilidad

Soportabilidad	Adaptabilidad
	Asistencia de Usuario
	Cambiabilidad
	Instalabilidad
	Mantenibilidad
	Testeabilidad

Usabilidad	Amigabilidad
	Asistencia de Usuario
	Atractivo
	Auto-Descripción
	Claridad
	Claridad
	Compleitud

	Comunicabilidad
	Confiabilidad
	Conformidad
	Consistencia
	Documentación
	Eficiencia
	Comprensibilidad
	Facilidad de aprendizaje
	Ingeniería humana
	Operatividad
	Personalización

Eficiencia	Abstracto
	Accesibilidad
	Cohesión
	Comportamiento en el tiempo
	Composición
	Conformidad
	Consumo de Recursos
	Encapsulamiento
	Herencia
	Polimorfismo

Interoperabilidad	Conformidad
	Modularidad

Corrección	Compleitud
	Consistencia
	Trazabilidad

Mantenibilidad	Abstracto
	Analizabilidad
	Auto-Descripción
	Cambiabilidad
	Cohesión
	Compleitud
	Conformidad
	Concisión
	Consistencia
	Documentación

	Encapsulamiento
	Comprensibilidad
	Estabilidad
	Modularidad
	Reusabilidad
	Simplicidad
	Testeabilidad

Testeabilidad	Accesibilidad
	Auto-Descripción
	Comunicabilidad
	Conformidad
	Instrumentación
	Modularidad
	Simplicidad

Reusabilidad	Abstracto
	Auto-Descripción
	Cohesión
	Consistencia
	Documentación
	Encapsulamiento
	Generalidad
	Independencia de Hardware
	Independencia de Software
	Modularidad
	Tamaño

Cambiabilidad	Abstracto
	Herencia
	Polimorfismo

Comprensibilidad	Auto-Descripción
	Cohesión
	Concisión
	Encapsulamiento
	Legibilidad

Ingeniería humana	Accesibilidad
	Comunicabilidad

ANEXO K: RECOMENDACIONES POR DISCIPLINA

Las recomendaciones surgen a partir dos aspectos: por un lado, la evaluación de una (o un conjunto) de métricas y, por otro lado, la orientación que prestan las heurísticas, todas las anteriores son originadas desde la aplicación de un atributo de calidad, o un conjunto de ellos, a un producto software.

Las recomendaciones están inspiradas en las heurísticas propuestas por autores indicados en el anexo U, tales como, Bashar [48], Garzas [57], McConell [66] y Riel [82]. Una recomendación puede estar sustentada en más de una heurística, propuestas por los anteriores autores.

Para una mejor comprensión, las recomendaciones están agrupadas por disciplina. A continuación están la totalidad de recomendaciones que surgen a partir de tres elementos principales: atributos, métricas y heurísticas.

ANÁLISIS

RECOMENDACIÓN 1

Los casos de uso demasiado cortos suelen ser incompletos o no representan realmente una interacción actor–sistema resultando generalmente triviales. Por otro lado, los casos de uso demasiado largos son frecuentemente incomprensibles, presentan un nivel de detalle excesivo, siendo más difíciles de leer. Un rango recomendado de pasos al interior de caso de uso debería ser entre 4 y 9 pasos.

RECOMENDACIÓN 2

Tenga presente el hecho de que un caso de uso sirve básicamente para expresar una interacción actor–sistema. Por ello, el número de pasos de actor y el de pasos de sistema deben estar en torno al 50%, considerando también la posibilidad de que existan pasos de inclusión o extensión que los realice otro caso de uso. El porcentaje de pasos recomendado a ejecutar en el caso de uso debería estar entre el 30 y 60 por ciento.

RECOMENDACIÓN 3

En situaciones en las que la proporción de pasos de sistema, es decir, la cantidad de pasos de sistema respecto al total de pasos en el caso de uso, sea alto (superior al 80%), el defecto más habitual es que se pretende representar un proceso *de sistema* que no necesita participación de ningún actor, por lo que no es realmente un caso de uso. Es por esto que es recomendable expresar el requisito mediante uno funcional redactado en texto plano, sin usar la estructura de caso de uso. Algunos casos de uso abstractos con una proporción de pasos de sistema inferior a 40%, no presentan defectos porque, al ser un caso de uso abstracto un fragmento de otros casos de uso, puede ocurrir que los pasos de actor estén descritos en los caso(s) de uso que lo activan. Si el caso de uso es concreto y pretende representarse una interacción actor–sistema, recomienda incluir al menos un paso de actor, el primero, algo que no tiene porqué ser cierto en los casos de uso abstractos.

RECOMENDACIÓN 4

Por lo general, los casos de uso describen interacciones entre actores y el sistema. Recomienda que un caso de uso tenga al menos 2 pasos para una interacción mínima *actor requiere – sistema responde*. Por tanto, verifique la completitud potencial de aquellos casos de uso con número de pasos inferior a 2. De otro lado, un valor muy alto del número de pasos (por ejemplo 20) es un

indicador de excesiva Complejidad o de la carencia de estructura (i.e. relaciones *include* o *extend*) en casos de uso.

RECOMENDACIÓN 5

La aproximación inicial para identificar alternativas de diseño, es una aproximación con un enfoque sobre la identificación de objetos sintéticos y del mundo real. Recomienda seguir los siguientes pasos para diseñar con objetos:

- Identificar los objetos y sus atributos (métodos y datos)
- Determinar que puede ser realizado por cada objeto.
- Determinar las partes de cada objeto que será visible a otros objetos---qué partes serán públicas y privadas.
- Definir la interfaz pública de cada objeto.

DISEÑO

RECOMENDACIÓN 6

Es recomendable considerar la similitud de datos, comportamiento y/o interfaces tan grande como sea posible en la jerarquía de Herencia, teniendo en cuenta que todas las clases abstractas deben ser clases base y además todas las clases base deberían ser clases abstractas.

RECOMENDACIÓN 7

Se sugiere conservar el número de funciones miembro de una clase en una cantidad manejable tan bien como el número de atributos al interior de la misma, tal que la clase tenga sólo las responsabilidades necesarias.

RECOMENDACIÓN 8

Tenga cuidado de clases que tienen muchos métodos de acceso definidos en su interfaz pública. Tener mucho sugiere que los datos relacionados y el comportamiento no están siendo conservados en un lugar.

RECOMENDACIÓN 9

Se recomienda tener cuidado de clases que tienen mucho comportamiento sin comunicación, esto es, métodos que operan sobre un subconjunto propio de datos miembro de una clase. Las clases Dios a menudo exhiben gran trato de comportamiento sin comunicación. Esto con el objeto de eliminar clases irrelevantes del sistema y también aquellas que estén fuera del sistema.

RECOMENDACIÓN 10

Una tarea clave en el diseño de una clase es decidir cuáles características deberían ser conocidas fuera de la clase y cuales de las restantes permanecerán ocultas. Una clase debería usar 25 rutinas y exponer solamente 5 de ellas, usando las otras 20 internamente. Recomienda que, dada una clase, ésta debería usar varios tipos de datos y no exponer información alguna acerca de ellos. Este aspecto del diseño de clases es conocido como “visibilidad” desde lo que esta tiene que hacer con aquellas características de la clase que están “visibles” o “expuestas” al exterior de la clase. Esto evita manejar clases grandes y poco cohesionadas.

RECOMENDACIÓN 11

Comúnmente, una clase capturaría una y solo una Abstracción clave con toda su información y todo su comportamiento. Una entidad principal al interior de un modelo de dominio es conocido como Abstracción clave. Recomienda que, si el modelo tiene más de una Abstracción, éste debería involucrar más de una clase. Algunas veces es difícil determinar todas las clases afectadas. A menudo algunas de esas clases son muy pequeñas y pueden ser eliminadas. Hay una posibilidad que el diseñador ha capturado que cada función es a una clase. Combinando métodos y atributos en una única clase debería resultar en una clase muy grande, aún cuando sería un modelo de Abstracción correcto.

RECOMENDACIÓN 12

Se recomienda que, a un nivel más detallado, pensar de cada interfaz de clase como un contrato con el resto del programa puede conllevar buenas ideas. Típicamente, el contrato es algo como “si promete darme los datos X, Y, Z y usted promete que tendrán las características A, B y C, prometo desarrollar las operaciones 1, 2 y 3 con las restricciones 8, 9 y 10”. Los contratos son usables para el manejo de la Complejidad porque, al menos en teoría, el objeto puede seguramente ignorar cualquier comportamiento no contractual. En la práctica, este aspecto es mucho más difícil.

RECOMENDACIÓN 13

Un raciocinio que puede generar ideas de diseños interesantes es preguntar a qué parecerá el sistema si lo diseña para facilitar pruebas. Es por ello que se sugiere dar respuesta a estos interrogantes: ¿Necesita separar la interfaz de usuario del resto del código para que pueda ejercitar independientemente? ¿Necesita organizar cada subsistema tal que minimice dependencias de otros subsistemas? El diseño para pruebas tiende a resultar en interfaces de clase más formalizadas, que generalmente es benéfico.

RECOMENDACIÓN 14

La meta que sugiere la modularidad es hacer cada rutina o clase como una “caja negra”. Usted sabe qué ingresa, y sabe qué sale, pero no sabe qué sucede al interior. Una caja negra tiene tanto una interfaz simple como una Funcionalidad bien definida que para alguna entrada específica puede predecir adecuadamente la salida correspondiente. Si sus rutinas son como cajas negras, entonces son perfectamente modulares, desarrollan acciones bien definidas y tienen interfaces simples. El concepto de modularidad relacionado con el ocultamiento de información, Encapsulamiento y otros conceptos de diseño. Tenga en cuenta que, algunas veces pensando en cómo ensamblar un sistema desde un conjunto de cajas negras, provee ideas que el ocultamiento de información y el Encapsulamiento no dan.

RECOMENDACIÓN 15

Se recomienda mantener todos los datos ocultos en su clase, a fin de conservar relacionados datos y comportamiento en un mismo lugar.

RECOMENDACIÓN 16

Es aconsejable que las clases deberían sólo exhibir nulo o exportar acoplamiento hacia otras clases, esto es, una clase debería solo usar operaciones en la interfaz pública de otra clase o tener nada que hacer con esa clase, minimizando así el número de mensajes enviados entre una clase y sus colaboradores.

RECOMENDACIÓN 17

Una clase abstracta es una *interfaz* reutilizable. Representa un conjunto de obligaciones de diseño funcionales y no funcionales que deben satisfacerse por un objeto concreto, poblando una instancia de esa clase. Una clase abstracta es típicamente un producto de un dominio de ingeniería. Está construido cuando un análisis de dominio identifica la necesidad por un componente que represente una interfaz común y procesamiento de Abstracción. Es por esto que recomienda tener presente que algunas veces requiera una implementación separada para el contexto puesto ahí, a fin de evitar problemas de solapamiento de clases.

RECOMENDACIÓN 18

Las clases derivadas deben tener conocimiento de sus clases base por definición, pero las clases base deberían no saber cualquier cosa sobre sus clases derivadas. Recuerde que la Herencia debería ser usada solamente para modelar una jerarquía de especialización.

RECOMENDACIÓN 19

Se recomienda que, si dos o más clases tienen datos y comportamiento común (i.e., métodos), entonces aquellas clases deberían heredar de una clase base común que capture aquellos datos y comportamiento. Pero, si dos o más clases comparten solamente una interfaz común (i.e., mensajes, no métodos), entonces deberían heredar de una clase base común solamente si serán usadas aplicando Polimorfismo. De otro lado, si dos o más clases comparten solamente datos comunes (no comportamiento común), entonces esos datos comunes deberían estar puestos en una clase que estará contenida por cada una de las clases compartidas.

RECOMENDACIÓN 20

No modele la semántica dinámica de una clase a través del uso de una relación de Herencia. Un intento para modelar la semántica dinámica con una relación semántica estática le guiará a una alternancia de tipos en tiempo de ejecución. Por lo tanto, recomienda que, si piensa que necesita crear nuevas clases en tiempo de ejecución, dé un paso atrás para darse cuenta que lo que está tratando de crear son objetos. Ahora generalice aquellos objetos en una clase.

RECOMENDACIÓN 21

Cada vez que ha encontrado una relación de Herencia múltiple en un diseño orientado a objetos, asegúrese que ninguna clase base es actualmente una clase derivada de otra clase base. Aún así, si tiene un ejemplo de Herencia múltiple en su diseño, asuma que tiene un error y luego pruebe otra forma.

RECOMENDACIÓN 22

Se sugiere a los diseñadores orientados a objetos que no deberían permitir criterios de diseño físico que corrompan sus diseños lógicos. Sin embargo, los criterios de diseño físico a menudo son usados en el proceso de elaboración de decisiones en tiempo de diseño lógico.

RECOMENDACIÓN 23

Teniendo muchos mensajes entre clases haciendo mucho o métodos que tienen mala granularidad aumentan la Complejidad del sistema a mantener. Con grandes interfaces públicas, también para los diseñadores es difícil encontrar un método requerido fácilmente, haciendo la clase difícil de entender y más compleja. Una clase que captura más de un conjunto de abstracciones sugiere que debería ser eliminada. Cabe anotar en este aspecto que la mayoría de métodos de clase deberían utilizar la mayoría de atributos de clase.

RECOMENDACIÓN 24

Se sugiere que las raíces de un árbol de Herencia sean abstractas. Desde la clase raíz, es posiblemente la más importante que provee una interfaz. Las clases bases abstractas que distribuyen la implementación se convierten en problemáticas en tanto el tamaño de la jerarquía de Herencia incrementa. Entonces la raíz de la jerarquía de Herencia debería ser una definición de tipo abstracto.

RECOMENDACIÓN 25

Es recomendable que una clase capture una y solamente una Abstracción clave con toda su información y todo su comportamiento, teniendo en cuenta que cada atributo debería estar oculto en su clase, buscando no crear clases innecesarias para modelar roles y teniendo en cuenta que, si hay propiedades comunes de instancias, éstas deberían estar definidas en una localización única.

RECOMENDACIÓN 26

Respecto a la Herencia, recomienda:

Limitar el número de agregados definidos por el usuario.

Restringir el acceso directo a agregados por el usuario.

Limite el uso de Herencia múltiple.

La jerarquía de Herencia debería no ser más de seis en profundidad, buscando Estabilidad al descender entre la jerarquía.

PRUEBAS

RECOMENDACIÓN 27

La mayor información que tendremos, la más inteligente que probaremos. Los probadores deberían ser capaces de entender bien el diseño, los cambios en el diseño y las dependencias entre componentes internos, externos y compartidos. La documentación técnica debería ser accesible al instante, precisa, bien organizada, específica y detallada, tal que permita hacer eficiente las actividades de prueba del sistema.

RECOMENDACIÓN 28

La cantidad de errores debería ser reducida al mínimo, tal que ninguno de ellos permita bloquear la ejecución de pruebas y, a su vez, la Evolución del sistema hacia otros estados funcionales (en simultánea al desarrollo y pruebas del sistema)

RECOMENDACIÓN 29

Un control adecuado de las tareas que realiza el software facilita y agiliza un manejo adecuado del mecanismo de pruebas.

Verifique que, en lo posible:

Todas las salidas puedan ser generadas y el código pueda ser ejecutado a través de una combinación de entradas.

Los estados de software y hardware puedan ser controlados directamente por el ingeniero de pruebas.

Los formatos de entradas y salidas sean consistentes y Estructurados.

La(s) prueba(a) a aplicar estén lo suficientemente especificadas, claras y exactas para ser replicadas en el sistema.

RECOMENDACIÓN 30

A través del control de pruebas, podemos rápidamente aislar problemas y desarrollar pruebas eficientes y efectivas. El sistema software debería ser construido desde módulos independientes que puedan ser probados independientemente. Los puntos a considerar en este respecto son simplicidad funcional (e.g. conjunto mínimo de características), estructural (e.g., modularización de la arquitectura) y de código (e.g., la adopción de un estándar de codificación).

RECOMENDACIÓN 31

Se recomienda que, aunque el cambio es inherente al software, poca cantidad de ellos implicaría pocos “trastornos” a probar. De aquí que, los cambios en el software deberían no ser tan frecuentes, aunque controlados de manera que no invaliden pruebas existentes, teniendo en cuenta que el software debería ser capaz de recuperarse bien ante fallas.

RECOMENDACIÓN 32

Se recomienda que el personal de pruebas tenga la mayor información sobre el software a probar, de manera que sea capaz de entender bien el diseño, los cambios en el diseño y las dependencias entre componentes internos, externos y compartidos. La documentación técnica debería ser accesible al instante, precisa, bien organizada, específica y detallada.

MANTENIMIENTO

RECOMENDACIÓN 33

Se recomienda que, para una jerarquía de Herencia, las dependencias de la clase descendiente deberían ser una consideración principal en la elaboración de cualquier decisión de refactorización.

ANÁLISIS Y DISEÑO

RECOMENDACIÓN 34

Se recomienda que, para lograr un alto nivel de Abstracción, haga uso de diagramas como una herramienta heurística poderosa que logra buenas representaciones de gran variedad de aspectos en diferentes instancias del transcurso del proyecto. Recuerde que una imagen vale más que mil palabras.

RECOMENDACIÓN 35

El análisis de casos explícito sobre un tipo de un objeto es usualmente un error. Recomienda al diseñador que haga uso de Polimorfismo en la mayoría de los casos, teniendo en cuenta que una clase debería capturar una y solo una Abstracción clave.

RECOMENDACIÓN 36

Es aconsejable eliminar dependencias desde clases de base de datos a sus clientes a fin de evitar acoplamiento y cohesión innecesarios.

DISEÑO E IMPLEMENTACIÓN

RECOMENDACIÓN 37

En ocasiones, se recomienda que aquellas propiedades comunes podrían ser una Abstracción única, una clase y una instancia. Esto hace la propiedad más centralizada y reutilizable.

RECOMENDACIÓN 38

Se recomienda no colocar detalles de implementación tales como funciones privadas de código compartido en la interfaz pública de una clase, a menos que sea necesario.

RECOMENDACIÓN 39

Se recomienda no crear clases u objetos Dios en su sistema. Sospeche de una clase cuyo nombre contenga Manejador, Director, Sistema, o Subsistema.

RECOMENDACIÓN 40

Las clases no deberían contener más objetos que un desarrollador pueda adaptar en su memoria de corto plazo. Un valor recomendado para éste es el número 6.

RECOMENDACIÓN 41

Se recomienda que aquella información semántica sobre la cual una restricción esta basada es mejor descentralizarla entre las clases involucradas en la restricción cuando la información es estable, pero es mejor centralizarla solamente si no está en manos de terceros; los objetos de terceros tienen ésta información volátil.

RECOMENDACIÓN 42

Se recomienda manejar como privados los datos en una clase base; no usar datos protegidos a menos que la jerarquía lo amerite.

RECOMENDACIÓN 43

Se recomienda que, en la práctica, las jerarquías de Herencia no sean tan profundas tal que una persona promedio pueda conservarla en su memoria de corto tiempo, de manera que sea fácil de mantener. Un valor común para esta profundidad es seis.

RECOMENDACIÓN 44

Evite depender de elementos que puedan ser cambiados cuando encuentre relaciones en una jerarquía de Herencia, esto sugiere que las dependencias de clases concretas lo sean pero de clases abstractas.

RECOMENDACIÓN 45

Se recomienda que, cuando haga uso de servicios y éstos tengan muchos parámetros, en lo posible evítelos, de modo que reduzca el acoplamiento y aumente la Comprensibilidad de su función en el sistema.

DISEÑO Y MANTENIMIENTO

RECOMENDACIÓN 46

Se recomienda que las propiedades de la clase permanezcan ocultas detrás de métodos de acceso. Esto significa que las propiedades de la clase no deberían ser usadas en la interfaz pública de una clase. La interfaz pública de una clase es una lista de métodos y atributos que podrían ser utilizados desde el exterior de la clase.

RECOMENDACIÓN 47

Se sugiere evite el uso de interfaces que soporten a muchos clientes con distintas necesidades, tal que una interfaz solamente maneje lo realmente necesario para un cliente.

PRUEBAS Y MANTENIMIENTO

RECOMENDACIÓN 48

Buscando consistencia en el sistema, sugiere que éste debe prevenir que los usuarios tengan que preguntarse si las diversas palabras, situaciones, o acciones significan la misma cosa. La consistencia puede darse de las siguientes maneras:

Visual: Los elementos de la interfaz de usuario deben ser consecuentes o consistentes en el aspecto y estructura.

Funcional: La manera de llevar a cabo las diferentes tareas a través del sistema debe ser consecuente, con otros sistemas similares, e incluso entre diferente clases de aplicaciones en el mismo sistema.

Evolutiva: Debe ser fácil tener una nueva versión del software, y que esta sea consistente con las anteriores. Como por ejemplo, en el caso de una familia de producto software, la consistencia sobre los productos en la familia es un aspecto importante.

RECOMENDACIÓN 49

Los seres humanos tienen limitaciones cognitivas del tratamiento de la información en memoria a corto plazo, por ello sugiere que lo que muestra por pantalla sea simple, por ejemplo presentar siete ítems por pantalla es una sobrecarga de información porque según estudios realizados siete u ocho es el medio límite de memoria a corto plazo. De igual forma sugiere que el usuario no debe tener que recordar la información de una parte de diálogo a la otra; es mejor mantener objetos, acciones, y opciones visibles que memorizar.

RECOMENDACIÓN 50

Se sugiere que el sistema debe proveer una relación clara entre lo que el usuario quiere hacer y el mecanismo para hacerlo. Esta propiedad puede estar estructurada de la siguiente manera:

Predecible: Para el usuario el comportamiento del sistema debe ser predecible.

Lenguaje de los usuarios: El sistema debe hablar el lenguaje de los usuarios, con las palabras, las frases y los conceptos familiares, en lugar de que los términos estén orientados al sistema. Utilizar convenciones del mundo real, haciendo que la información aparezca en un orden natural y lógico.

Facilidad de la navegación: Debe ser obvio al usuario cómo navegar en el sistema.

RECOMENDACIÓN 51

Se recomienda que un componente sea juzgado como completamente correcto, cuando es puesto en un contexto y sometido a pruebas de tiempo y cronograma en un espacio de análisis. Sin embargo, es posible verificar que un componente es correcto con respecto a su propia implementación.

ANEXO L: SONDEO DE EMPRESAS DE DESARROLLO DE SOFTWARE EN EL SUR OCCIDENTE COLOMBIANO

A continuación presentamos los casos de estudio con los que los resultados de la aplicación del marco conceptual en un conjunto de MIPyMESPS son mostrados. Para ello, estructuramos la información de la siguiente manera:

Se realizó una primera visita, en la que es obtenido un primer acercamiento a las empresas a partir de una encuesta que permita conocer aspectos clave de las empresas para el desarrollo del trabajo.

L1. Formato de la encuesta

El formato de dicha encuesta es el siguiente:

Encuesta: Sondeo de empresas de desarrollo de software en el sur occidente Colombiano

Proyecto: Marco Conceptual De Atributos, Métricas, y Heurísticas de Calidad Para La Valoración Del Producto Software Orientado a Objetos.

Institución: Universidad del Cauca (Popayán-Colombia)

Responsables:

MAG. Jorge Jair Moreno Chaustre

Estudiantes: Liliam Paola Bolaños Rengifo
Manuel Alejandro Navia Porras

Confidencialidad: Los responsables de la encuesta aquí propuesta se comprometen a no difundir la información personal o comercial suministrada por el encuestado; ésta tendrá uso anónimo, salvo si las partes deciden lo contrario. Los resultados y datos estadísticos generales obtenidos de la información que arroje esta encuesta serán utilizados en el proyecto mencionado aquí.

Objetivo: A través esta encuesta se pretende conocer empresas de desarrollo de software, del sur occidente Colombiano, y tener un acercamiento a la dinámica de las mismas en relación a los productos software.

DATOS DEL ENTREVISTADO	
Organización:	
Nombre Completo:	
Correos Electrónicos:	
Ciudad:	
Fecha:	

Marque con una X:

1. Indique cual es el tiempo de existencia de la empresa : _____
2. Volumen de ventas por año: _____
3. Número de empleados de la empresa: _____
4. Número de empleados dedicados al desarrollo de software: _____

5. **¿Qué tipo de productos desarrolla en su empresa?**

Tipo	Clase
<input type="checkbox"/> Aplicaciones web <input type="checkbox"/> Aplicaciones de escritorio <input type="checkbox"/> Aplicaciones distribuidas <input type="checkbox"/> Aplicaciones móviles <input type="checkbox"/> Otra (s) <hr/>	<input type="checkbox"/> Sistemas Contables <input type="checkbox"/> Software Educativo <input type="checkbox"/> Software para la salud <input type="checkbox"/> Sistemas de información <input type="checkbox"/> Otra (s) <hr/>

6. **¿Cuál es el alcance interior (zonas próximas de operación) de la empresa? :**

7. **¿Cuál es el alcance exterior (zonas lejanas de operación) de la empresa?:**

8. **Indique el número de proyectos en curso (iniciados):** _____

9. **Indique el número de proyectos terminados:** _____

10. **Indique el número de proyectos cancelados:** _____

11. **Indique el promedio de duración por proyecto (tiempo):** _____

12. **¿Realiza pruebas de software?:**

Sí

No

13. **¿Existe equipo SQA para realizar pruebas?**

Sí

No

14. **¿Aplica métodos de validación o verificación para el producto?**

Sí

No

15. **¿Existe proceso de registro de pruebas?**

Sí

No

16. **Escriba sus comentarios y sugerencias (opcional).**

Gracias por su atención y colaboración.

Firma del entrevistado

Firma del entrevistador

L2. Resultados de la empresa QHATU

DATOS DEL ENTREVISTADO	
Organización:	QHATU
Nombre Completo:	ÁLVARO OJEDA BUCHELI
Correos Electrónicos:	ALVAROJEDA@COLOMBIA.COM
Ciudad:	PASTO
Fecha:	MARZO17 2009

Marque con una X:

- Indique cual es el tiempo de existencia de la empresa : _____2 AÑOS_____
- Volumen de ventas por año: _____20 MILLONES_____
- Número de empleados de la empresa: _____5_____
- Número de empleados dedicados al desarrollo de software: _____5_____
- ¿Qué tipo de productos desarrolla en su empresa?

Tipo	Clase
___ Aplicaciones Web	_X_ Sistemas Contables
___ Aplicaciones de escritorio	___ Software Educativo
___ Aplicaciones distribuidas	___ Software para la salud
___ Aplicaciones móviles	___ Sistemas de información
___ Otra (s) _____	___ Otra (s) _____

- ¿Cuál es el alcance interior (zonas próximas de operación) de la empresa? :
_____NARIÑO_____
- ¿Cuál es el alcance exterior (zonas lejanas de operación) de la empresa?:
_____CAUCA_____
- Indique el número de proyectos en curso (iniciados): _____1_____
- Indique el número de proyectos terminados: _____6_____
- Indique el número de proyectos cancelados: _____1_____
- Indique el promedio de duración por proyecto (tiempo): _____3 MESES_____
- ¿Realiza pruebas de software?:
Sí X
No
- ¿Existe equipo SQA para realizar pruebas?
Sí
No X
- ¿Aplica métodos de validación o verificación para el producto?

Sí X
 No

15. ¿Existe proceso de registro de pruebas?

Sí
 No X

16. Escriba sus comentarios y sugerencias (opcional).

L3. Resultados empresa Latin Business

DATOS DEL ENTREVISTADO	
Organización:	LATIN-BUSINESS
Nombre Completo:	John Jairo Ortiz Canchala
Correos Electrónicos:	jjortiz@parquesoftpasto.com
Ciudad:	Pasto
Fecha:	marzo 11 de 2009

Marque con una X:

- Indique cual es el tiempo de existencia de la empresa : __5 años__
- Volumen de ventas por año: ____10 millones__
- Número de empleados de la empresa: __3 personas__
- Número de empleados dedicados al desarrollo de software: __2 personas__
- ¿Qué tipo de productos desarrolla en su empresa?

Tipo	Clase
<input checked="" type="checkbox"/> Aplicaciones web	<input type="checkbox"/> Sistemas Contables
<input type="checkbox"/> Aplicaciones de escritorio	<input type="checkbox"/> Software Educativo
<input type="checkbox"/> Aplicaciones distribuidas	<input checked="" type="checkbox"/> Software para la salud
<input type="checkbox"/> Aplicaciones móviles	<input checked="" type="checkbox"/> Sistemas de información
<input type="checkbox"/> Otra (s) _____	<input type="checkbox"/> Otra (s) _____

- ¿Cuál es el alcance interior (zonas próximas de operación) de la empresa? :
 Nariño _____
- ¿Cuál es el alcance exterior (zonas lejanas de operación) de la empresa?:
 Colombia _____
- Indique el número de proyectos en curso (iniciados): __2__
- Indique el número de proyectos terminados: __1__

10. Indique el número de proyectos cancelados: 2
11. Indique el promedio de duración por proyecto (tiempo): 3 meses
12. ¿Realiza pruebas de software?:
 Sí
 No
13. ¿Existe equipo SQA para realizar pruebas?
 Sí
 No
14. ¿Aplica métodos de validación o verificación para el producto?
 Sí
 No
15. ¿Existe proceso de registro de pruebas?
 Sí
 No
16. Escriba sus comentarios y sugerencias (opcional).

L4. Resultados empresa SITI

DATOS DEL ENTREVISTADO	
Organización:	Soluciones Integrales en Tecnologías de la información
Nombre Completo:	Gelber Moran Silva
Correos Electrónicos:	gelbermoran@siticol.com
Ciudad:	Pasto
Fecha:	marzo 13 de 2009

Marque con una X:

1. Indique cual es el tiempo de existencia de la empresa : 7 años
2. Volumen de ventas por año: 50 millones
3. Número de empleados de la empresa: 3 personas
4. Número de empleados dedicados al desarrollo de software: 2 personas
5. ¿Qué tipo de productos desarrolla en su empresa?

Tipo	Clase
------	-------

<input checked="" type="checkbox"/> Aplicaciones Web <input type="checkbox"/> Aplicaciones de escritorio <input type="checkbox"/> Aplicaciones distribuidas <input type="checkbox"/> Aplicaciones móviles <input type="checkbox"/> Otra (s) _____ 	<input type="checkbox"/> Sistemas Contables <input checked="" type="checkbox"/> Software Educativo <input type="checkbox"/> Software para la salud <input checked="" type="checkbox"/> Sistemas de información <input type="checkbox"/> Otra (s) __Servicios de comunicaciones Web__
---	--

6. ¿Cuál es el alcance interior (zonas próximas de operación) de la empresa? :
 _____ Pasto, Túquerres, Ipiales _____
7. ¿Cuál es el alcance exterior (zonas lejanas de operación) de la empresa?:
 _____ Putumayo (Mocoa), Bogotá, Ibarra (Ecuador) _____
8. Indique el número de proyectos en curso (iniciados): _____ 3 _____
9. Indique el número de proyectos terminados: _____ 7 _____
10. Indique el número de proyectos cancelados: _____ 0 _____
11. Indique el promedio de duración por proyecto (tiempo): _____ 6 meses _____
12. ¿Realiza pruebas de software?:
 Sí
 No
13. ¿Existe equipo SQA para realizar pruebas?
 Sí
 No
14. ¿Aplica métodos de validación o verificación para el producto?
 Sí
 No
15. ¿Existe proceso de registro de pruebas?
 Sí
 No
16. Escriba sus comentarios y sugerencias (opcional).

L5. Resultados empresa SERATIC

DATOS DEL ENTREVISTADO	
Organización:	SERATIC LTDA
Nombre Completo:	Diego Iván Chamorro
Correos Electrónicos:	diego.chamorro@seratic.com
Ciudad:	Popayán
Fecha:	Febrero 26 de 2009

Marque con una X:

1. **Indique cual es el tiempo de existencia de la empresa :** ____ 4 años _____
2. **Volumen de ventas por año:** ____ 220 millones _____
3. **Número de empleados de la empresa:** ____ 9 personas _____
4. **Número de empleados dedicados al desarrollo de software:** ____ 7 personas _____
5. **¿Qué tipo de productos desarrolla en su empresa?**

Tipo	Clase
<input checked="" type="checkbox"/> Aplicaciones Web	<input type="checkbox"/> Sistemas Contables
<input type="checkbox"/> Aplicaciones de escritorio	<input type="checkbox"/> Software Educativo
<input type="checkbox"/> Aplicaciones distribuidas	<input type="checkbox"/> Software para la salud
<input checked="" type="checkbox"/> Aplicaciones móviles	<input type="checkbox"/> Sistemas de información
<input type="checkbox"/> Otra (s)	<input type="checkbox"/> Otra (s)
_____	_____

6. **¿Cuál es el alcance interior (zonas próximas de operación) de la empresa? :**
____ Valle, Cundinamarca _____
7. **¿Cuál es el alcance exterior (zonas lejanas de operación) de la empresa?:**
____ Peru _____
8. **Indique el número de proyectos en curso (iniciados):** ____ 1 _____
9. **Indique el número de proyectos terminados:** ____ 4 _____
10. **Indique el número de proyectos cancelados:** ____ 0 _____
11. **Indique el promedio de duración por proyecto (tiempo):** ____ 3 meses _____
12. **¿Realiza pruebas de software?:**
 Sí
 No
13. **¿Existe equipo SQA para realizar pruebas?**
 Sí
 No
14. **¿Aplica métodos de validación o verificación para el producto?**
 Sí
 No
15. **¿Existe proceso de registro de pruebas?**
 Sí
 No
16. **Escriba sus comentarios y sugerencias (opcional).**

L6. Resultados empresa INPUT

DATOS DEL ENTREVISTADO	
Organización:	INPUT Technologies Ltda.
Nombre Completo:	Ricardo Andrés Ledezma
Correos Electrónicos:	ricardo.ledezma@input.com.co , rledezma@hotmail.com
Ciudad:	Popayán
Fecha:	Marzo 5 de 2009

Marque con una X:

1. Indique cual es el tiempo de existencia de la empresa : __3 años__
2. Volumen de ventas por año: ____15 millones__
3. Número de empleados de la empresa: ____3__
4. Número de empleados dedicados al desarrollo de software: __3__
5. ¿Qué tipo de productos desarrolla en su empresa?

Tipo	Clase
<input checked="" type="checkbox"/> Aplicaciones Web	<input type="checkbox"/> Sistemas Contables
<input type="checkbox"/> Aplicaciones de escritorio	<input type="checkbox"/> Software Educativo
<input type="checkbox"/> Aplicaciones distribuidas	<input type="checkbox"/> Software para la salud
<input type="checkbox"/> Aplicaciones móviles	<input checked="" type="checkbox"/> Sistemas de información
<input type="checkbox"/> Otra (s)	<input type="checkbox"/> Otra (s)

6. ¿Cuál es el alcance interior (zonas próximas de operación) de la empresa? :
____Popayán, sur occidente colombiano____
7. ¿Cuál es el alcance exterior (zonas lejanas de operación) de la empresa?:
____Colombia, Ecuador____
8. Indique el número de proyectos en curso (iniciados): __4__
9. Indique el número de proyectos terminados: ____3__
10. Indique el número de proyectos cancelados: ____0__
11. Indique el promedio de duración por proyecto (tiempo): __6 meses__
12. ¿Realiza pruebas de software?:
Sí
No
13. ¿Existe equipo SQA para realizar pruebas?

Sí
No

14. **¿Aplica métodos de validación o verificación para el producto?**

Sí
No

15. **¿Existe proceso de registro de pruebas?**

Sí
No

16. **Escriba sus comentarios y sugerencias (opcional).**

L7. Resultados generales

Algunos resultados que son percibidos en la encuesta son los siguientes:

- El tiempo promedio de existencia (desde su creación hasta el momento) de las empresas es de 4 años, esto en relación con el carácter de las empresas (MiPyMESPS) y las características por las cuales surgieron de acuerdo a las condiciones de creación de las mismas.
- El número de empleados promedio es de 5, lo cual es congruente con el carácter de las empresas consultadas. Así mismo, el número de empleados promedio dedicados al desarrollo de software es de 4.
- El tipo de desarrollo de software al cual están dedicadas las empresas es, en su mayoría, el de las aplicaciones Web enfocados principalmente en sistemas de información, con menor participación en áreas como sistemas contables, software educativo, entre otros.
- Con respecto al alcance geográfico para la distribución de sus productos, en general, cada una ha gestado su mercado tanto regional, nacional e internacional, lo cual es un buen incentivo para el crecimiento de las empresas de desarrollo de software.
- Aunque ninguna de las empresas encuestadas cuenta con equipo de SQA, todas indican realizar pruebas de software, registrando los resultados generados tras aplicar métodos de verificación o validación a los productos evaluados. Sin embargo, manifiestan también que no realiza un proceso formal de pruebas.

ANEXO M: ENCUESTA PARA EL DIAGNOSTICO RESPECTO A PRACTICAS DEL PRODUCTO SOFTWARE

A continuación pueden observarse los casos de estudio con los que son mostrados los resultados de la aplicación del marco conceptual en un conjunto de MIPyMESPS. Para ello, estructuramos la información de la siguiente manera:

Se realizó una primera visita, en la que es obtenido un primer acercamiento a las empresas a partir de una encuesta que permita conocer aspectos clave de las empresas para el desarrollo del trabajo.

M1. Formato de la encuesta

El formato de dicha encuesta es el siguiente:

Encuesta para el diagnóstico respecto a prácticas del producto software

Proyecto: Marco Conceptual De Atributos, Métricas, y Heurísticas de Calidad Para La Valoración Del Producto Software Orientado a Objetos.

Institución: Universidad del Cauca (Popayán-Colombia)

Responsables:

MAG. Jorge Jair Moreno Chaustre

Estudiantes: Liliam Paola Bolaños Rengifo
Manuel Alejandro Navia Porras

Confidencialidad: Los responsables de la encuesta aquí propuesta se comprometen a no difundir la información personal o comercial suministrada por el encuestado; ésta tendrá uso anónimo, salvo si las partes deciden lo contrario. Los resultados y datos estadísticos generales obtenidos de la información que arroje esta encuesta serán utilizados en el proyecto mencionado aquí.

Objetivo: Conocer las prácticas relacionadas con el uso de métricas orientadas hacia la valoración del producto software. La siguiente encuesta pretende diagnosticar el modo de medición que se puede realizar a los productos software por las empresas del sur-occidente y proyectos universitarios dedicados al desarrollo de software.

DATOS DEL ENTREVISTADO	
Organización:	
Nombre Completo:	
Correos Electrónicos:	
Ciudad:	
Fecha:	

Marque con una X en el recuadro según corresponda:

1. ¿Sabe usted qué es una métrica de software?

Sí
No

2. ¿Utiliza métricas para valorar la calidad de su producto software?

Sí
No

3. **¿Conoce de la existencia de herramientas que calculan o extraen métricas de de software a través del código fuente?**

Sí
No

Si las conoce, ¿las utiliza?

Sí ¿Cual? _____
No

4. **Si utiliza las métricas de software, ¿cómo las calcula?**

- Manualmente
 Por medio de herramienta(s)
 Utiliza medición manual y por medio de herramienta(s)

5. **Si utiliza la medición manual para lo anterior, entonces:**

a. Describa brevemente el proceso de medición:

b. **¿Qué métricas de calidad de software utiliza? (puede seleccionar más de una)**

- | | |
|---|---|
| <input type="checkbox"/> Métricas de cobertura de pruebas | <input type="checkbox"/> Métricas de legibilidad |
| <input type="checkbox"/> Métricas de comprensión del programa | <input type="checkbox"/> Métricas de tamaño |
| <input type="checkbox"/> Métricas de puntos de función | <input type="checkbox"/> Métricas de líneas de código |
| <input type="checkbox"/> Otra (s) _____ | |

6. **Si utiliza herramienta(s) para la medición de calidad del producto software, ¿Cuál(es) usa?**

- | | |
|---|--------------------------------------|
| <input type="checkbox"/> Visual Studio Team Sistema | <input type="checkbox"/> FxCop |
| <input type="checkbox"/> Rational Suite | <input type="checkbox"/> ToroMetrics |
| <input type="checkbox"/> SDMETRCIS | <input type="checkbox"/> Kemis |
| <input type="checkbox"/> MetricView | <input type="checkbox"/> Anaconda |
| <input type="checkbox"/> Otra (s) _____ | |

7. **En sus desarrollos, ¿conoce y Aplica el paradigma orientado a objetos?**

- Lo conoce y lo Aplica No lo conoce
 Lo conoce pero no lo Aplica

8. **¿Aplica alguna arquitectura software en sus proyectos? (Puede seleccionar más de una opción)**

- En Capas No Aplica arquitectura
 Orientada a Objetos
 Aplica Otra (s) _____

9. **¿Qué técnicas o estrategias utiliza para detectar los errores en su producto?**

- | | |
|---|--|
| <input type="checkbox"/> Pruebas de unidad | <input type="checkbox"/> Pruebas de sistema |
| <input type="checkbox"/> Pruebas de integración | <input type="checkbox"/> Pruebas de validación |
| <input type="checkbox"/> Pruebas de aceptación | <input type="checkbox"/> |
| <input type="checkbox"/> Otra (s) _____ | |

10. **Escriba sus comentarios y sugerencias (opcional).**

Gracias por su atención y colaboración.

Firma del entrevistado

Firma del entrevistador

M2. Resultados de la empresa QHATU

DATOS DEL ENTREVISTADO	
Organización:	QHATU
Nombre Completo:	ÁLVARO OJEDA BUCHELI
Correos Electrónicos:	ALVAROJEDA@COLOMBIA.COM
Ciudad:	PASTO
Fecha:	MARZO 17 DE 2009

Marque con una X en el recuadro según corresponda:

1. ¿Sabe usted qué es una métrica de software?

Sí

No (Continúe con la pregunta 7)

2. ¿Utiliza métricas para valorar la calidad de su producto software?

Sí

No

3. ¿Conoce de la existencia de herramientas que calculan o extraen métricas de de software a través del código fuente?

Sí

No

Si las conoce, ¿las utiliza?

Sí ¿Cual? _____

No

4. Si utiliza las métricas de software, ¿cómo las calcula?

Manualmente

Por medio de herramienta(s)

Utiliza medición manual y por medio de herramienta(s)

5. Si utiliza la medición manual para lo anterior, entonces:

a. Describa brevemente el proceso de medición:

b. ¿Qué métricas de calidad de software utiliza? (puede seleccionar más de una)

Métricas de cobertura de pruebas

Métricas de legibilidad

Métricas de comprensión del programa

Métricas de tamaño

Métricas de puntos de función

Métricas de líneas de código

Otra (s) _____

6. Si utiliza herramienta(s) para la medición de calidad del producto software, ¿Cuál(es) usa?

- | | |
|---|--------------------------------------|
| <input type="checkbox"/> Visual Studio Team Sistema | <input type="checkbox"/> FxCop |
| <input type="checkbox"/> Rational Suite | <input type="checkbox"/> ToroMetrics |
| <input type="checkbox"/> SDMETRCIS | <input type="checkbox"/> Kemis |
| <input type="checkbox"/> MetricView | <input type="checkbox"/> Anaconda |
| <input type="checkbox"/> Otra (s) _____ | |

7. En sus desarrollos, ¿conoce y Aplica el paradigma orientado a objetos?

- Lo conoce y lo Aplica No lo conoce
 Lo conoce pero no lo Aplica

8. ¿Aplica alguna arquitectura software en sus proyectos? (Puede seleccionar más de una opción)

- En Capas No Aplica arquitectura
 Orientada a Objetos
 Aplica Otra (s) _____ CLIENTE SERVIDOR _____

9. ¿Qué técnicas o estrategias utiliza para detectar los errores en su producto?

- | | |
|--|---|
| <input type="checkbox"/> Pruebas de unidad | <input checked="" type="checkbox"/> Pruebas de sistema |
| <input checked="" type="checkbox"/> Pruebas de integración | <input checked="" type="checkbox"/> Pruebas de validación |
| <input checked="" type="checkbox"/> Pruebas de aceptación | <input type="checkbox"/> |
| <input type="checkbox"/> Otra (s) _____ | |

10. Escriba sus comentarios y sugerencias (opcional).

_____ EN LA PREGUNTA 10 LA PREGUNTA 7 NO EXISTE, SE PRESUME QUE DEBE SER LA NUMERO 17 _____

M3. Resultados empresa Latin Business

DATOS DEL ENTREVISTADO	
Organización:	LATIN-BUSINESS
Nombre Completo:	John Jairo Ortiz Canchala
Correos Electrónicos:	jjortiz@parquesoftpasto.com
Ciudad:	Pasto
Fecha:	marzo 11 de 2009

Marque con una X en el recuadro según corresponda:

1. ¿Sabe usted qué es una métrica de software?

- Sí
No

2. ¿Utiliza métricas para valorar la calidad de su producto software?

- Sí
No

3. ¿Conoce de la existencia de herramientas que calculan o extraen métricas de de software a través del código fuente?

- Sí

No

Si las conoce, ¿las utiliza?

Sí ¿Cual? _____

No

4. Si utiliza las métricas de software, ¿cómo las calcula?

Manualmente

Por medio de herramienta(s)

Utiliza medición manual y por medio de herramienta(s)

5. Si utiliza la medición manual para lo anterior, entonces:

a. Describa brevemente el proceso de medición:

b. ¿Qué métricas de calidad de software utiliza? (puede seleccionar más de una)

Métricas de cobertura de pruebas

Métricas de legibilidad

Métricas de comprensión del programa

Métricas de tamaño

Métricas de puntos de función

Métricas de líneas de código

Otra (s) _____

6. Si utiliza herramienta(s) para la medición de calidad del producto software, ¿Cuál(es) usa?

Visual Studio Team Sistema

FxCop

Rational Suite

ToroMetrics

SDMETRCIS

Kemis

MetricView

Anaconda

Otra (s) _____

7. En sus desarrollos, ¿conoce y Aplica el paradigma orientado a objetos?

Lo conoce y lo Aplica

No lo conoce

Lo conoce pero no lo Aplica

8. ¿Aplica alguna arquitectura software en sus proyectos? (Puede seleccionar más de una opción)

En Capas

No Aplica arquitectura

Orientada a Objetos

Aplica Otra (s) _____

9. ¿Qué técnicas o estrategias utiliza para detectar los errores en su producto?

Pruebas de unidad

Pruebas de sistema

Pruebas de integración

Pruebas de validación

Pruebas de aceptación

Otra (s) _____

10. Escriba sus comentarios y sugerencias (opcional).

M4. Resultados empresa SITI

DATOS DEL ENTREVISTADO	
Organización:	Soluciones Integrales en Tecnologías de la información
Nombre Completo:	Gelber Moran Silva
Correos Electrónicos:	gelbermoran@siticol.com
Ciudad:	Pasto
Fecha:	marzo 13 de 2009

Marque con una X en el recuadro según corresponda:

1. ¿Sabe usted qué es una métrica de software?

Sí
 No

2. ¿Utiliza métricas para valorar la calidad de su producto software?

Sí
 No

3. ¿Conoce de la existencia de herramientas que calculan o extraen métricas de de software a través del código fuente?

Sí
 No

Si las conoce, ¿las utiliza?

Sí ¿Cual? _____
 No

4. Si utiliza las métricas de software, ¿cómo las calcula?

- Manualmente
 Por medio de herramienta(s)
 Utiliza medición manual y por medio de herramienta(s)

5. Si utiliza la medición manual para lo anterior, entonces:

a. Describa brevemente el proceso de medición:

b. ¿Qué métricas de calidad de software utiliza? (puede seleccionar más de una)

- | | |
|---|---|
| <input type="checkbox"/> Métricas de cobertura de pruebas | <input type="checkbox"/> Métricas de legibilidad |
| <input type="checkbox"/> Métricas de comprensión del programa | <input type="checkbox"/> Métricas de tamaño |
| <input type="checkbox"/> Métricas de puntos de función | <input type="checkbox"/> Métricas de líneas de código |
| <input type="checkbox"/> Otra (s) _____ | |

6. Si utiliza herramienta(s) para la medición de calidad del producto software, ¿Cuál(es) usa?

- | | |
|---|--------------------------------------|
| <input type="checkbox"/> Visual Studio Team Sistema | <input type="checkbox"/> FxCop |
| <input type="checkbox"/> Rational Suite | <input type="checkbox"/> ToroMetrics |
| <input type="checkbox"/> SDMETRCIS | <input type="checkbox"/> Kemis |
| <input type="checkbox"/> MetricView | <input type="checkbox"/> Anaconda |

Otra (s) _____

7. En sus desarrollos, ¿conoce y Aplica el paradigma orientado a objetos?
 Lo conoce y lo Aplica No lo conoce
 Lo conoce pero no lo Aplica

8. ¿Aplica alguna arquitectura software en sus proyectos? (Puede seleccionar más de una opción)
 En Capas No Aplica arquitectura
 Orientada a Objetos
 Aplica Otra (s) _____

9. ¿Qué técnicas o estrategias utiliza para detectar los errores en su producto?
 Pruebas de unidad Pruebas de sistema
 Pruebas de integración Pruebas de validación
 Pruebas de aceptación
 Otra (s) _____

10. Escriba sus comentarios y sugerencias (opcional).

M5. Resultados empresa SERATIC

DATOS DEL ENTREVISTADO	
Organización:	SERATIC LTDA
Nombre Completo:	Diego Iván Chamorro
Correos Electrónicos:	diego.chamorro@seratic.com
Ciudad:	Popayán
Fecha:	Febrero 26 de 2009

Marque con una X en el recuadro según corresponda:

1. ¿Sabe usted qué es una métrica de software?
Sí
No
2. ¿Utiliza métricas para valorar la calidad de su producto software?
Sí
No
3. ¿Conoce de la existencia de herramientas que calculan o extraen métricas de de software a través del código fuente?
Sí
No
- Si las conoce, ¿las utiliza?
Sí ¿Cual? _____
No

4. Si utiliza las métricas de software, ¿cómo las calcula?

- Manualmente
 Por medio de herramienta(s)
 Utiliza medición manual y por medio de herramienta(s)

5. Si utiliza la medición manual para lo anterior, entonces:

a. Describa brevemente el proceso de medición:

b. ¿Qué métricas de calidad de software utiliza? (puede seleccionar más de una)

- | | |
|---|---|
| <input type="checkbox"/> Métricas de cobertura de pruebas | <input type="checkbox"/> Métricas de legibilidad |
| <input type="checkbox"/> Métricas de comprensión del programa | <input type="checkbox"/> Métricas de tamaño |
| <input type="checkbox"/> Métricas de puntos de función | <input type="checkbox"/> Métricas de líneas de código |
| <input type="checkbox"/> Otra (s) _____ | |

6. Si utiliza herramienta(s) para la medición de calidad del producto software, ¿Cuál(es) usa?

- | | |
|---|--------------------------------------|
| <input type="checkbox"/> Visual Studio Team Sistema | <input type="checkbox"/> FxCop |
| <input type="checkbox"/> Rational Suite | <input type="checkbox"/> ToroMetrics |
| <input type="checkbox"/> SDMETRCIS | <input type="checkbox"/> Kemis |
| <input type="checkbox"/> MetricView | <input type="checkbox"/> Anaconda |
| <input type="checkbox"/> Otra (s) _____ | |

7. En sus desarrollos, ¿conoce y Aplica el paradigma orientado a objetos?

- Lo conoce y lo Aplica No lo conoce
 Lo conoce pero no lo Aplica

8. ¿Aplica alguna arquitectura software en sus proyectos? (Puede seleccionar más de una opción)

- En Capas No Aplica arquitectura
 Orientada a Objetos
 Aplica Otra (s) _____

9. ¿Qué técnicas o estrategias utiliza para detectar los errores en su producto?

- | | |
|--|---|
| <input checked="" type="checkbox"/> Pruebas de unidad | <input type="checkbox"/> Pruebas de sistema |
| <input checked="" type="checkbox"/> Pruebas de integración | <input checked="" type="checkbox"/> Pruebas de validación |
| <input type="checkbox"/> Pruebas de aceptación | <input type="checkbox"/> |
| <input type="checkbox"/> Otra (s) _____ | |

10. Escriba sus comentarios y sugerencias (opcional).

M6. Resultados empresa INPUT

DATOS DEL ENTREVISTADO	
Organización:	INPUT Technologies Ltda.
Nombre Completo:	Ricardo Andrés Ledezma

Correos Electrónicos:	ricardo.ledezma@input.com.co , rledezma@hotmail.com
Ciudad:	Popayán
Fecha:	Marzo 5 de 2009

Marque con una X en el recuadro según corresponda:

1. ¿Sabe usted qué es una métrica de software?

Sí
 No

2. ¿Utiliza métricas para valorar la calidad de su producto software?

Sí
 No

3. ¿Conoce de la existencia de herramientas que calculan o extraen métricas de de software a través del código fuente?

Sí
 No

Si las conoce, ¿las utiliza?

Sí ¿Cual? _____
 No

4. Si utiliza las métricas de software, ¿cómo las calcula?

- Manualmente
 Por medio de herramienta(s)
 Utiliza medición manual y por medio de herramienta(s)

5. Si utiliza la medición manual para lo anterior, entonces:

a. Describa brevemente el proceso de medición:

b. ¿Qué métricas de calidad de software utiliza? (puede seleccionar más de una)

- | | |
|---|---|
| <input type="checkbox"/> Métricas de cobertura de pruebas | <input type="checkbox"/> Métricas de legibilidad |
| <input type="checkbox"/> Métricas de comprensión del programa | <input type="checkbox"/> Métricas de tamaño |
| <input type="checkbox"/> Métricas de puntos de función | <input type="checkbox"/> Métricas de líneas de código |
| <input type="checkbox"/> Otra (s) _____ | |

6. Si utiliza herramienta(s) para la medición de calidad del producto software, ¿Cuál(es) usa?

- | | |
|---|--------------------------------------|
| <input type="checkbox"/> Visual Studio Team Sistema | <input type="checkbox"/> FxCop |
| <input type="checkbox"/> Rational Suite | <input type="checkbox"/> ToroMetrics |
| <input type="checkbox"/> SDMETRCIS | <input type="checkbox"/> Kemis |
| <input type="checkbox"/> MetricView | <input type="checkbox"/> Anaconda |
| <input type="checkbox"/> Otra (s) _____ | |

7. En sus desarrollos, ¿conoce y Aplica el paradigma orientado a objetos?

Lo conoce y lo Aplica
 Lo conoce pero no lo Aplica
 No lo conoce

8. ¿Aplica alguna arquitectura software en sus proyectos? (Puede seleccionar más de una)

opción)

- En Capas No Aplica arquitectura
 Orientada a Objetos
 Aplica Otra (s) _cliente-servidor_____

9. ¿Qué técnicas o estrategias utiliza para detectar los errores en su producto?

- Pruebas de unidad Pruebas de sistema
 Pruebas de integración Pruebas de validación
 Pruebas de aceptación
 Otra (s) _____

10. Escriba sus comentarios y sugerencias (opcional).

M7. Resultados generales

Algunos de los resultados percibidos en la encuesta son los siguientes:

- Un porcentaje considerable de los encuestados (85%) afirman no saber qué es una métrica de software, por lo que no es posible indagar más con respecto a la medición de calidad de software a través de éstos elementos, en las empresas que no tenían conocimiento respecto al tema de mención.
- Aquellas personas que afirman saber qué es una métrica de software, no la utilizan para la valoración de calidad de software de sus productos o para cualquier otro fin; además, a pesar de conocer de la existencia de herramientas que calculan o extraen métricas de software a través del código fuente, no hace uso de ellas.
- Al indagar sobre el paradigma de programación utilizado, la totalidad de los encuestados afirma conocer y aplicar el orientado a objetos, como también la mayoría afirman aplicar arquitectura de software en capas, aunque no exclusivamente, es decir que no son regidos todo el tiempo por una arquitectura en particular sino que eligen de acuerdo al tipo de software que esperan obtener. Es así como, para algunos casos, utilizan la arquitectura cliente-servidor u orientada a objetos.
- Entre las técnicas o estrategias más usadas por los encuestados para detectar los errores en los productos software que desarrollan están, en primer lugar están las pruebas de validación, en segundo lugar están las pruebas de aceptación (80% de los encuestados) y en tercer lugar están las pruebas de unidad, de integración y de sistema.

ANEXO N: ENCUESTA: INSPECCIÓN DE LAS PRÁCTICAS APLICADAS AL PRODUCTO SOFTWARE

A continuación pueden observarse los casos de estudio con los que son mostrados los resultados de la aplicación del marco conceptual en un conjunto de MIPyMESPS. Para ello, estructuramos la información de la siguiente manera:

Se realizó una primera visita, en la que son obtenidos un primer acercamiento a las empresas a partir de una encuesta que permita conocer aspectos clave de las empresas para el desarrollo del trabajo.

N1. Formato de la encuesta

El formato de dicha encuesta es el siguiente:

Encuesta: Inspección de prácticas aplicadas al producto software

Proyecto: Marco Conceptual De Atributos, Métricas, y Heurísticas de Calidad Para La Valoración Del Producto Software Orientado a Objetos.

Institución: Universidad del Cauca (Popayán-Colombia)

Responsables:

MAG. Jorge Jair Moreno Chaustre

Estudiantes:

Liliam Paola Bolaños Rengifo

Manuel Alejandro Navia Porras

Confidencialidad: Los responsables de la encuesta aquí propuesta se comprometen a no difundir la información personal o comercial suministrada por el encuestado; ésta tendrá uso anónimo, salvo si las partes deciden lo contrario. Los resultados y datos estadísticos generales obtenidos de la información que arroje esta encuesta serán utilizados en el proyecto mencionado aquí.

Objetivo: Inspeccionar sobre prácticas aplicadas para la valoración del producto software en la región.

La siguiente encuesta pretende indagar si en la región se realiza medición de la calidad de software, por las empresas dedicadas al desarrollo de software en el sur-occidente Colombiano.

DATOS DEL ENTREVISTADO	
Organización:	
Nombre Completo:	
Correos Electrónicos:	
Ciudad:	
Fecha:	

Marque con una X en el recuadro según corresponda:

7. En sus desarrollos con respecto al paradigma oo, indique lo siguiente:

- Lo conoce y lo Aplica
 Lo conoce pero no lo Aplica
 No lo conoce

8. ¿Aplica alguna arquitectura software en sus proyectos? (Puede seleccionar más de una opción)

- En Capas
 Orientada a Objetos
 No Aplica arquitectura
 Aplica Otra (s) _____

9. ¿En sus proyectos software hace uso de Procesos de desarrollo de software? (Puede seleccionar más de una opción)

- Cascada
 Espiral
 Prototipos
 RUP
 No hace uso de proceso
 Utiliza Otro (s) _____
- RAD
 Scrum
 XP
 Agile RUP

10. ¿En sus proyectos de software realiza medición para asegurar la calidad del mismo?

- Sí
No

11. Para sus productos de software, el porcentaje dedicado a la calidad es:

- En tiempo: 0% 1 - 11 - 31 - 51 - Mas del
10% 30% 50% 70% 70%
- En costos: 0% 1 - 11 - 31 - 51 - Mas del
10% 30% 50% 70% 70%
- En esfuerzo: 0% 1 - 11 - 31 - 51 - Mas del
10% 30% 50% 70% 70%

12. ¿Utiliza algún modelo de calidad de software para apoyar al proceso de desarrollo en sus proyectos?

- Sí ¿Cuál? _____
No

13. ¿Utiliza algún estándar de calidad de software para apoyar al proceso de desarrollo en sus proyectos?

- Sí ¿Cuál? _____
No

14. ¿Posee algún tipo de certificación de calidad de software?

- Sí ¿Cuál? _____
No

15. Escriba sus comentarios y sugerencias (opcional).

Gracias por su atención y colaboración.

Firma del entrevistado

Firma del entrevistador

N2. Resultados de la empresa QHATU

DATOS DEL ENTREVISTADO	
Organización:	QHATU
Nombre Completo:	ALVARO OJEDA BUCHELI
Correos Electrónicos:	ALVAROJEDA@COLOMBIA.COM
Ciudad:	PASTO
Fecha:	17 DE MARZO DE 2009

Marque con una X en el recuadro según corresponda:

1. **En sus desarrollos con respecto al paradigma oo, indique lo siguiente:**

- Lo conoce y lo Aplica
 Lo conoce pero no lo Aplica
 No lo conoce

2. **¿Aplica alguna arquitectura software en sus proyectos? (Puede seleccionar más de una opción)**

- En Capas
 Orientada a Objetos
 No Aplica arquitectura
 Aplica Otra (s) _____ CLIENTE SERVIDOR _____

3. **¿En sus proyectos software hace uso de Procesos de desarrollo de software? (Puede seleccionar más de una opción)**

- | | |
|---|------------------------------------|
| <input type="checkbox"/> Cascada | <input type="checkbox"/> RAD |
| <input checked="" type="checkbox"/> Espiral | <input type="checkbox"/> Scrum |
| <input checked="" type="checkbox"/> Prototipos | <input type="checkbox"/> XP |
| <input type="checkbox"/> RUP | <input type="checkbox"/> Agile RUP |
| <input type="checkbox"/> No hace uso de proceso | |
| <input type="checkbox"/> Utiliza Otro (s) _____ | |

4. ¿En sus proyectos de software realiza medición para asegurar la calidad del mismo?

Sí
 No

5. Para sus productos de software, el porcentaje dedicado a la calidad es:

En tiempo: 0% 1 - 11 - 31 - 51 - Mas del
 10% 30% 50% 70% 70%

En costos: 0% 1 - 11 - 31 - 51 - Mas del
 10% 30% 50% 70% 70%

En esfuerzo: 0% 1 - 11 - 31 - 51 - Mas del
 10% 30% 50% 70% 70%

6. ¿Utiliza algún modelo de calidad de software para apoyar al proceso de desarrollo en sus proyectos?

Sí ¿Cuál? _____
 No

7. ¿Utiliza algún estándar de calidad de software para apoyar al proceso de desarrollo en sus proyectos?

Sí ¿Cuál? _____
 No

8. ¿Posee algún tipo de certificación de calidad de software?

Sí ¿Cuál? _____
 No

9. Escriba sus comentarios y sugerencias (opcional).

N3. Resultados empresa Latin Business

DATOS DEL ENTREVISTADO	
Organización:	LATIN-BUSINESS
Nombre Completo:	John Jairo Ortiz Canchala
Correos Electrónicos:	jjortiz@parquesoftpasto.com
Ciudad:	Pasto
Fecha:	marzo 11 de 2009

Marque con una X en el recuadro según corresponda:

1. En sus desarrollos con respecto al paradigma oo, indique lo siguiente:

Lo conoce y lo Aplica

- Lo conoce pero no lo Aplica
- No lo conoce

2. ¿Aplica alguna arquitectura software en sus proyectos? (Puede seleccionar más de una opción)

- En Capas
- Orientada a Objetos
- No Aplica arquitectura
- Aplica Otra (s) _____

3. ¿En sus proyectos software hace uso de Procesos de desarrollo de software? (Puede seleccionar más de una opción)

- Cascada
- Espiral
- Prototipos
- RUP
- No hace uso de proceso
- Utiliza Otro (s) _____
- RAD
- Scrum
- XP
- Agile RUP

4. ¿En sus proyectos de software realiza medición para asegurar la calidad del mismo?

- Sí
- No

5. Para sus productos de software, el porcentaje dedicado a la calidad es:

- En tiempo:** 0% 10% - 11-30% - 31-50% - 51-70% - Mas del 70%
- En costos:** 0% 10% - 11-30% - 31-50% - 51-70% - Mas del 70%
- En esfuerzo:** 0% 10% - 11-30% - 31-50% - 51-70% - Mas del 70%

6. ¿Utiliza algún modelo de calidad de software para apoyar al proceso de desarrollo en sus proyectos?

- Sí ¿Cuál? _____
- No

7. ¿Utiliza algún estándar de calidad de software para apoyar al proceso de desarrollo en sus proyectos?

- Sí ¿Cuál? _____
- No

8. ¿Posee algún tipo de certificación de calidad de software?

- Sí ¿Cuál? _____
- No

9. Escriba sus comentarios y sugerencias (opcional).

N4. Resultados empresa SITI

DATOS DEL ENTREVISTADO	
Organización:	Soluciones Integrales en Tecnologías de la información
Nombre Completo:	Gelber Moran Silva
Correos Electrónicos:	gelbermoran@siticol.com
Ciudad:	Pasto
Fecha:	marzo 13 de 2009

Marque con una X en el recuadro según corresponda:

1. En sus desarrollos con respecto al paradigma oo, indique lo siguiente:

- Lo conoce y lo Aplica
 Lo conoce pero no lo Aplica
 No lo conoce

2. ¿Aplica alguna arquitectura software en sus proyectos? (Puede seleccionar más de una opción)

- En Capas
 Orientada a Objetos
 No Aplica arquitectura
 Aplica Otra (s) _____

3. ¿En sus proyectos software hace uso de Procesos de desarrollo de software? (Puede seleccionar más de una opción)

- Cascada RAD
 Espiral Scrum
 Prototipos XP
 RUP Agile RUP
 No hace uso de proceso
 Utiliza Otro (s) _____

4. ¿En sus proyectos de software realiza medición para asegurar la calidad del mismo?

- Sí
No

5. Para sus productos de software, el porcentaje dedicado a la calidad es:

- En tiempo:** 0% 1 - 11 - 31 - 51 - Mas del 70% del

En costos: 0% 10% - 11 30% - 31 50% - 51 70% - Mas del 70%

En esfuerzo: 0% 10% - 11 30% - 31 50% - 51 70% - Mas del 70%

6. ¿Utiliza algún modelo de calidad de software para apoyar al proceso de desarrollo en sus proyectos?

Sí ¿Cuál? _____

No

7. ¿Utiliza algún estándar de calidad de software para apoyar al proceso de desarrollo en sus proyectos?

Sí ¿Cuál? _____

No

8. ¿Posee algún tipo de certificación de calidad de software?

Sí ¿Cuál? _____

No

9. Escriba sus comentarios y sugerencias (opcional).

N5. Resultados empresa SERATIC

DATOS DEL ENTREVISTADO	
Organización:	SERATIC LTDA
Nombre Completo:	Diego Iván Chamorro
Correos Electrónicos:	diego.chamorro@seratic.com
Ciudad:	Popayán
Fecha:	Febrero 26 de 2009

Marque con una X en el recuadro según corresponda:

1. En sus desarrollos con respecto al paradigma oo, indique lo siguiente:

Lo conoce y lo Aplica

Lo conoce pero no lo Aplica

No lo conoce

2. ¿Aplica alguna arquitectura software en sus proyectos? (Puede seleccionar más de una opción)

En Capas

Orientada a Objetos

No Aplica arquitectura

Aplica Otra (s) _____

3. ¿En sus proyectos software hace uso de Procesos de desarrollo de software?

(Puede seleccionar más de una opción)

- Cascada RAD
 Espiral Scrum
 Prototipos XP
 RUP Agile RUP
 No hace uso de proceso
 Utiliza Otro (s) __implementando en la empresa y validando con COMPETISOFT__

4. ¿En sus proyectos de software realiza medición para asegurar la calidad del mismo?

Sí
No

5. Para sus productos de software, el porcentaje dedicado a la calidad es:

En tiempo: 0% 1 - 11 - 31 - 51 - Mas del
10% 30% 50% 70% 70%

En costos: 0% 1 - 11 - 31 - 51 - Mas del
10% 30% 50% 70% 70%

En esfuerzo: 0% 1 - 11 - 31 - 51 - Mas del
10% 30% 50% 70% 70%

6. ¿Utiliza algún modelo de calidad de software para apoyar al proceso de desarrollo en sus proyectos?

Sí ¿Cuál? Metodología propia para el desarrollo de aplicaciones móviles____
No

7. ¿Utiliza algún estándar de calidad de software para apoyar al proceso de desarrollo en sus proyectos?

Sí ¿Cuál? _____
No

8. ¿Posee algún tipo de certificación de calidad de software?

Sí ¿Cuál? _____
No

9. Escriba sus comentarios y sugerencias (opcional).

N6. Resultados empresa INPUT

DATOS DEL ENTREVISTADO

Organización:	INPUT Technologies Ltda
Nombre Completo:	Ricardo Andrés Ledezma
Correos Electrónicos:	ricardo.ledezma@input.com.co , rledezma@hotmail.com
Ciudad:	Popayán
Fecha:	Marzo 5 de 2009

Marque con una X en el recuadro según corresponda:

1. En sus desarrollos con respecto al paradigma oo, indique lo siguiente:

- Lo conoce y lo Aplica
 Lo conoce pero no lo Aplica
 No lo conoce

2. ¿Aplica alguna arquitectura software en sus proyectos? (Puede seleccionar más de una opción)

- En Capas
 Orientada a Objetos
 No Aplica arquitectura
 Aplica Otra (s) __Cliente-Servidor_____

3. ¿En sus proyectos software hace uso de Procesos de desarrollo de software? (Puede seleccionar más de una opción)

- Cascada RAD
 Espiral Scrum
 Prototipos XP
 RUP Agile RUP
 No hace uso de proceso
 Utiliza Otro (s) _Proceso definido por la empresa (pendiente Estandarización)___

4. ¿En sus proyectos de software realiza medición para asegurar la calidad del mismo?

- Sí
No

5. Para sus productos de software, el porcentaje dedicado a la calidad es:

- En tiempo:** 0% 10% - 30% - 50% - 70% - Mas del
- En costos:** 0% 10% - 30% - 50% - 70% - Mas del
- En esfuerzo:** 0% 10% - 30% - 50% - 70% - Mas del

6. ¿Utiliza algún modelo de calidad de software para apoyar al proceso de desarrollo en sus proyectos?

- Sí ¿Cuál? _____
No

7. ¿Utiliza algún estándar de calidad de software para apoyar al proceso de desarrollo

en sus proyectos?

Sí ¿Cuál? _____
No

8. ¿Posee algún tipo de certificación de calidad de software?

Sí ¿Cuál? _____
No

9. Escriba sus comentarios y sugerencias (opcional).

N7. Resultados generales

Algunos de los resultados percibidos en la encuesta son los siguientes:

- A partir de preguntas anteriores, en relación con la aplicación del paradigma orientado a objetos y de arquitectura alguna, en los diferentes desarrollos de software en las empresas, indagamos ahora respecto al uso procesos de desarrollo de software en los proyectos de las empresas encuestadas, obteniendo que en algunos casos (23% de los encuestados) hacen uso metodologías de desarrollo “tradicionales” tales como *espiral* y *prototipos*, aunque otras han definido uno en particular para su empresa.
- Indagando sobre si realiza medición para asegurar la calidad de los proyectos software, sólo un 5% de los encuestados dio una respuesta afirmativa.
- El porcentaje en tiempo, costo y esfuerzo de trabajo en el desarrollo de los proyectos, dedicado a la calidad del mismo, en promedio, no es superior al 30%. Esto esta reflejado dado que gran parte de ellos no hace uso alguno de modelo o estándar de calidad de software, para apoyar el proceso de desarrollo de software en sus proyectos. Adicionalmente, ninguno de los encuestados afirma poseer certificación de calidad.

ANEXO O: ENCUESTA: INSPECCIÓN DE FUENTES USADAS POR LAS EMPRESAS DE LA REGIÓN PARA LA CALIDAD DEL PRODUCTO SOFTWARE

A continuación puede observarse los casos de estudio con los que son mostrados los resultados de la aplicación del marco conceptual en un conjunto de MIPyMESPS. Para ello, estructuramos la información de la siguiente manera:

Se realizó una primera visita, en la que son obtenidos un primer acercamiento a las empresas a partir de una encuesta que permita conocer aspectos clave de las empresas para el desarrollo del trabajo.

01. Formato de la encuesta

El formato de dicha encuesta es el siguiente:

Encuesta: Inspección de fuentes usadas por las empresas de la región para la calidad del producto SW

Proyecto: Marco Conceptual De Atributos, Métricas, y Heurísticas de Calidad Para La Valoración Del Producto Software Orientado a Objetos.

Institución: Universidad del Cauca (Popayán-Colombia)

Responsables:

MAG. Jorge Jair Moreno Chaustre

Estudiantes: Liliam Paola Bolaños Rengifo
Manuel Alejandro Navia Porras

Confidencialidad: Los responsables de la encuesta aquí propuesta se comprometen a no difundir la información personal o comercial suministrada por el encuestado; ésta tendrá uso anónimo, salvo si las partes deciden lo contrario. Los resultados y datos estadísticos generales obtenidos de la información que arroje esta encuesta serán utilizados en el proyecto mencionado aquí.

Objetivo: inspeccionar sobre la dinámica de las empresas de la región en cuanto a la mejora la calidad para los productos software. La siguiente encuesta pretende indagar sobre el uso de fuentes de información usadas en empresas del sur-occidente Colombiano, para mejorar la calidad de sus productos software.

DATOS DEL ENTREVISTADO	
Organización:	
Nombre Completo:	
Correos Electrónicos:	
Ciudad:	
Fecha:	

Marque con una X en el recuadro según corresponda:

1. ¿En sus desarrollos de software cuáles son las características de calidad que considera más apropiadas, tomando en cuenta el objetivo de uso del producto de software? (Puede seleccionar más de una opción)

<input type="checkbox"/> Funcionalidad	<input type="checkbox"/> Preciso
<input type="checkbox"/> Confiabilidad	<input type="checkbox"/> Consistente
<input type="checkbox"/> Usabilidad	<input type="checkbox"/> Computable
<input type="checkbox"/> Eficiencia	<input type="checkbox"/> Acoplamiento
<input type="checkbox"/> Mantenibilidad	<input type="checkbox"/> Cohesión
<input type="checkbox"/> Portabilidad	
<input type="checkbox"/> Utiliza	Otro

(s)

2. ¿De qué depende la selección de las características que tendrán los productos que desarrolla? (Puede seleccionar más de una opción)

<input type="checkbox"/> Requerimientos del cliente	<input type="checkbox"/> Modelos/estándares de calidad
<input type="checkbox"/> Metodología de desarrollo	<input type="checkbox"/> Arquitectura
<input type="checkbox"/> Alcance, tiempo y costo	
<input type="checkbox"/> Otro	

(s)

3. ¿Utiliza alguno de los siguientes Modelos/estándares?

<input type="checkbox"/> ISO 9126	<input type="checkbox"/> BOEHM
<input type="checkbox"/> IEEE 729	<input type="checkbox"/> DROMEY
<input type="checkbox"/> IEEE 610.12	<input type="checkbox"/> MCCALL
<input type="checkbox"/> ISO 14598	<input type="checkbox"/> BANSIYA
<input type="checkbox"/> FURPS	<input type="checkbox"/> ABAS
<input type="checkbox"/> Otro (s)	

4. ¿Cómo comprueba que su producto alcanza el nivel de calidad adecuado?

<input type="checkbox"/> Aceptación del cliente	<input type="checkbox"/> Funcionamiento del producto
<input type="checkbox"/> Certificación de calidad	
<input type="checkbox"/> Otro	

(s)

5. ¿Cuáles fuentes consulta o ha tenido en cuenta para consolidar un conjunto de buenas prácticas en el desarrollo de sus productos software? (escriba en el recuadro)

Internet	
Libros	
Documentos	
Estándares	
Otro(s)	

6. Escriba sus comentarios y sugerencias (opcional).

Gracias por su atención y colaboración.

Firma del entrevistado

Firma del entrevistador

02. Resultados de la empresa QHATU

DATOS DEL ENTREVISTADO	
Organización:	QHATU
Nombre Completo:	ALVARO OJEDA BUCHELI
Correos Electrónicos:	ALVAROJEDA@COLOMBIA.COM
Ciudad:	PASTO
Fecha:	17 MARZO DE 2009

Marque con una X en el recuadro según corresponda:

1. ¿En sus desarrollos de software cuales son las características de calidad que considera más apropiadas, tomando en cuenta el objetivo de uso del producto de software? (Puede seleccionar más de una opción)

- | | |
|---|---------------------------------------|
| <input checked="" type="checkbox"/> Funcionalidad | <input type="checkbox"/> Preciso |
| <input type="checkbox"/> Confiabilidad | <input type="checkbox"/> Consistente |
| <input checked="" type="checkbox"/> Usabilidad | <input type="checkbox"/> Computable |
| <input checked="" type="checkbox"/> Eficiencia | <input type="checkbox"/> Acoplamiento |
| <input type="checkbox"/> Mantenibilidad | <input type="checkbox"/> Cohesión |
| <input checked="" type="checkbox"/> Portabilidad | |
| <input type="checkbox"/> Utiliza Otro (s) _____ | |

2. ¿De qué depende la selección de las características que tendrán los productos que desarrolla? (Puede seleccionar más de una opción)

- | | |
|--|--|
| <input checked="" type="checkbox"/> Requerimientos del cliente | <input type="checkbox"/> Modelos/estándares de calidad |
| <input type="checkbox"/> Metodología de desarrollo | <input checked="" type="checkbox"/> Arquitectura |
| <input type="checkbox"/> Alcance, tiempo y costo | |
| <input type="checkbox"/> Otro(s) _____ | |

3. ¿Utiliza alguno de los siguientes Modelos/estándares?

- | | |
|---|----------------------------------|
| <input type="checkbox"/> ISO 9126 | <input type="checkbox"/> BOEHM |
| <input type="checkbox"/> IEEE 729 | <input type="checkbox"/> DROMEY |
| <input type="checkbox"/> IEEE 610.12 | <input type="checkbox"/> MCCALL |
| <input type="checkbox"/> ISO 14598 | <input type="checkbox"/> BANSIYA |
| <input type="checkbox"/> FURPS | <input type="checkbox"/> ABAS |
| <input type="checkbox"/> Otro (s) _____ | |

4. ¿Cómo comprueba que su producto alcanza el nivel de calidad adecuado?

- | | |
|--|---|
| <input checked="" type="checkbox"/> Aceptación del cliente | <input checked="" type="checkbox"/> Funcionamiento del producto |
| <input type="checkbox"/> Certificación de calidad | |
| <input type="checkbox"/> Otro(s) _____ | |

5. ¿Cuáles fuentes consulta o ha tenido en cuenta para consolidar un conjunto de buenas prácticas en el desarrollo de sus productos software? (escriba en el recuadro)

Internet	X
Libros	X

Documentos	X
Estándares	X
Otro(s)	

6. **Escriba sus comentarios y sugerencias (opcional).**

O3. Resultados empresa Latin Business

DATOS DEL ENTREVISTADO	
Organización:	LATIN-BUSINESS
Nombre Completo:	John Jairo Ortiz Canchala
Correos Electrónicos:	jjortiz@parquesoftpasto.com
Ciudad:	Pasto
Fecha:	marzo 11 de 2009

Marque con una X en el recuadro según corresponda:

1. **¿En sus desarrollos de software cuáles son las características de calidad que considera más apropiadas, tomando en cuenta el objetivo de uso del producto de software? (Puede seleccionar más de una opción)**

- | | |
|---|---------------------------------------|
| <input checked="" type="checkbox"/> Funcionalidad | <input type="checkbox"/> Preciso |
| <input checked="" type="checkbox"/> Confiabilidad | <input type="checkbox"/> Consistente |
| <input checked="" type="checkbox"/> Usabilidad | <input type="checkbox"/> Computable |
| <input type="checkbox"/> Eficiencia | <input type="checkbox"/> Acoplamiento |
| <input type="checkbox"/> Mantenibilidad | <input type="checkbox"/> Cohesión |
| <input checked="" type="checkbox"/> Portabilidad | |
| <input type="checkbox"/> Utiliza Otro (s) _____ | |

2. **¿De qué depende la selección de las características que tendrán los productos que desarrolla? (Puede seleccionar más de una opción)**

- | | |
|--|--|
| <input checked="" type="checkbox"/> Requerimientos del cliente | <input type="checkbox"/> Modelos/estándares de calidad |
| <input checked="" type="checkbox"/> Metodología de desarrollo | <input type="checkbox"/> Arquitectura |
| <input checked="" type="checkbox"/> Alcance, tiempo y costo | |
| <input type="checkbox"/> Otro (s) _____ | |

3. **¿Utiliza alguno de los siguientes Modelos/estándares?**

- | | |
|---|----------------------------------|
| <input type="checkbox"/> ISO 9126 | <input type="checkbox"/> BOEHM |
| <input type="checkbox"/> IEEE 729 | <input type="checkbox"/> DROMEY |
| <input type="checkbox"/> IEEE 610.12 | <input type="checkbox"/> MCCALL |
| <input type="checkbox"/> ISO 14598 | <input type="checkbox"/> BANSIYA |
| <input type="checkbox"/> FURPS | <input type="checkbox"/> ABAS |
| <input type="checkbox"/> Otro (s) _____ | |

4. **¿Cómo comprueba que su producto alcanza el nivel de calidad adecuado?**

- | | |
|--|--|
| <input checked="" type="checkbox"/> Aceptación del cliente | <input type="checkbox"/> Funcionamiento del producto |
|--|--|

- Certificación de calidad
 Otro (s) _____

5. ¿Cuáles fuentes consulta o ha tenido en cuenta para consolidar un conjunto de buenas prácticas en el desarrollo de sus productos software? (escriba en el recuadro)

Internet	X (Google)
Libros	
Documentos	
Estándares	
Otro(s)	

6. Escriba sus comentarios y sugerencias (opcional).

04. Resultados empresa SITI

DATOS DEL ENTREVISTADO	
Organización:	Soluciones Integrales en Tecnologías de la información
Nombre Completo:	Gelber Moran Silva
Correos Electrónicos:	gelbermoran@siticol.com
Ciudad:	Pasto
Fecha:	marzo 13 de 2009

Marque con una X en el recuadro según corresponda:

1. ¿En sus desarrollos de software cuáles son las características de calidad que considera más apropiadas, tomando en cuenta el objetivo de uso del producto de software? (Puede seleccionar más de una opción)

- | | |
|--|---------------------------------------|
| <input checked="" type="checkbox"/> Funcionalidad | <input type="checkbox"/> Preciso |
| <input type="checkbox"/> Confiabilidad | <input type="checkbox"/> Consistente |
| <input checked="" type="checkbox"/> Usabilidad | <input type="checkbox"/> Computable |
| <input checked="" type="checkbox"/> Eficiencia | <input type="checkbox"/> Acoplamiento |
| <input checked="" type="checkbox"/> Mantenibilidad | <input type="checkbox"/> Cohesión |
| <input type="checkbox"/> Portabilidad | |
| <input type="checkbox"/> Utiliza Otro (s) _____ | |

2. ¿De qué depende la selección de las características que tendrán los productos que desarrolla? (Puede seleccionar más de una opción)

- | | |
|--|--|
| <input checked="" type="checkbox"/> Requerimientos del cliente | <input type="checkbox"/> Modelos/estándares de calidad |
| <input checked="" type="checkbox"/> Metodología de desarrollo | <input checked="" type="checkbox"/> Arquitectura |
| <input checked="" type="checkbox"/> Alcance, tiempo y costo | |
| <input type="checkbox"/> Otro (s) _____ | |

3. ¿Utiliza alguno de los siguientes Modelos/estándares?

- | | |
|--------------------------------------|---------------------------------|
| <input type="checkbox"/> ISO 9126 | <input type="checkbox"/> BOEHM |
| <input type="checkbox"/> IEEE 729 | <input type="checkbox"/> DROMEY |
| <input type="checkbox"/> IEEE 610.12 | <input type="checkbox"/> MCCALL |

- ISO 14598 BANSIYA
 FURPS ABAS
 Otro (s) _____

4. ¿Cómo comprueba que su producto alcanza el nivel de calidad adecuado?

- Aceptación del cliente Funcionamiento del producto
 Certificación de calidad
 Otro (s) _____

5. ¿Cuáles fuentes consulta o ha tenido en cuenta para consolidar un conjunto de buenas prácticas en el desarrollo de sus productos software? (escriba en el recuadro)

Internet	X (desarrolloweb.com, Developer.yahoo.com)
Libros	
Documentos	
Estándares	
Otro(s)	

6. Escriba sus comentarios y sugerencias (opcional).

O5. Resultados empresa SERATIC

DATOS DEL ENTREVISTADO	
Organización:	SERATIC LTDA
Nombre Completo:	Diego Iván Chamorro
Correos Electrónicos:	diego.chamorro@seratic.com
Ciudad:	Popayán
Fecha:	Febrero 26 de 2009

Marque con una X en el recuadro según corresponda:

1. ¿En sus desarrollos de software cuáles son las características de calidad que considera más apropiadas, tomando en cuenta el objetivo de uso del producto de software? (Puede seleccionar más de una opción)

- Funcionalidad Preciso
 Confiabilidad Consistente
 Usabilidad Computable
 Eficiencia Acoplamiento
 Mantenibilidad Cohesión
 Portabilidad
 Utiliza Otro (s) _____

2. ¿De qué depende la selección de las características que tendrán los productos que desarrolla? (Puede seleccionar más de una opción)

- Requerimientos del cliente Modelos/estándares de calidad
 Metodología de desarrollo Arquitectura
 Alcance, tiempo y costo

Otro (s) _____

3. **¿Utiliza alguno de los siguientes Modelos/estándares?**

- | | |
|---|----------------------------------|
| <input type="checkbox"/> ISO 9126 | <input type="checkbox"/> BOEHM |
| <input type="checkbox"/> IEEE 729 | <input type="checkbox"/> DROMEY |
| <input type="checkbox"/> IEEE 610.12 | <input type="checkbox"/> MCCALL |
| <input type="checkbox"/> ISO 14598 | <input type="checkbox"/> BANSIYA |
| <input type="checkbox"/> FURPS | <input type="checkbox"/> ABAS |
| <input type="checkbox"/> Otro (s) _____ | |

4. **¿Cómo comprueba que su producto alcanza el nivel de calidad adecuado?**

- | | |
|--|---|
| <input checked="" type="checkbox"/> Aceptación del cliente | <input checked="" type="checkbox"/> Funcionamiento del producto |
| <input type="checkbox"/> Certificación de calidad | |
| <input type="checkbox"/> Otro (s) _____ | |

5. **¿Cuáles fuentes consulta o ha tenido en cuenta para consolidar un conjunto de buenas prácticas en el desarrollo de sus productos software? (escriba en el recuadro)**

Internet	
Libros	
Documentos	X (Metodología de referencia para el desarrollo de soluciones móviles)
Estándares	X (CMMI)
Otro(s)	X (Competisfot, experiencia con clientes)

6. **Escriba sus comentarios y sugerencias (opcional).**

06. Resultados empresa INPUT

DATOS DEL ENTREVISTADO	
Organización:	INPUT Technologies Ltda.
Nombre Completo:	Ricardo Andrés Ledezma
Correos Electrónicos:	ricardo.ledezma@input.com.co , rledezma@hotmail.com
Ciudad:	Popayán
Fecha:	Marzo 5 de 2009

Marque con una X en el recuadro según corresponda:

1. **¿En sus desarrollos de software cuáles son las características de calidad que considera más apropiadas, tomando en cuenta el objetivo de uso del producto de software? (Puede seleccionar más de una opción)**

- | | |
|---|---------------------------------------|
| <input checked="" type="checkbox"/> Funcionalidad | <input type="checkbox"/> Preciso |
| <input checked="" type="checkbox"/> Confiabilidad | <input type="checkbox"/> Consistente |
| <input checked="" type="checkbox"/> Usabilidad | <input type="checkbox"/> Computable |
| <input checked="" type="checkbox"/> Eficiencia | <input type="checkbox"/> Acoplamiento |
| <input type="checkbox"/> Mantenibilidad | <input type="checkbox"/> Cohesión |
| <input type="checkbox"/> Portabilidad | |
| <input type="checkbox"/> Utiliza Otro (s) _____ | |

2. ¿De qué depende la selección de las características que tendrán los productos que desarrolla? (Puede seleccionar más de una opción)

- Requerimientos del cliente Modelos/estándares de calidad
 Metodología de desarrollo Arquitectura
 Alcance, tiempo y costo
 Otro (s) _____

3. ¿Utiliza alguno de los siguientes Modelos/estándares?

- ISO 9126 BOEHM
 IEEE 729 DROMEY
 IEEE 610.12 MCCALL
 ISO 14598 BANSIYA
 FURPS ABAS
 Otro (s) _____

4. ¿Cómo comprueba que su producto alcanza el nivel de calidad adecuado?

- Aceptación del cliente Funcionamiento del producto
 Certificación de calidad
 Otro (s) _____

5. ¿Cuáles fuentes consulta o ha tenido en cuenta para consolidar un conjunto de buenas prácticas en el desarrollo de sus productos software? (escriba en el recuadro)

Internet	
Libros	
Documentos	X (Tesis: Marco de referencia centrado en la arquitectura para la mejora de características de Usabilidad)
Estándares	
Otro(s)	Competisoft

6. Escriba sus comentarios y sugerencias (opcional).

07. Resultados generales

Algunos de los resultados percibidos en la encuesta son los siguientes:

- Al consultar sobre las características de calidad consideradas como las más apropiadas, tomando en cuenta el objetivo de uso del producto software, la totalidad de los encuestados busca Funcionalidad y Usabilidad, y no menos importante es considerada la Portabilidad (75%), Eficiencia, Mantenibilidad y Confiabilidad (45%). Características como la Cohesión, el acoplamiento y la consistencia no fueron tenidas en cuenta por los encuestados.

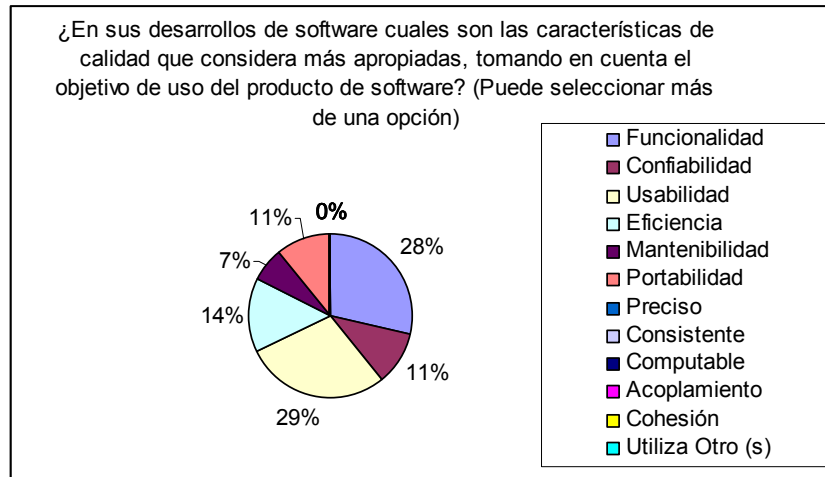


Gráfico 1. Características de calidad elegidas por los encuestados

- En consecuencia, la selección de características que tienen los productos software que desarrollan depende, para la totalidad de los encuestados, de los requerimientos a satisfacer en cada producto software. No menos importante es considerada la arquitectura software, el alcance, tiempo y costo necesario para la obtención del producto (80% de los encuestados) y la metodología de desarrollo (60% de los encuestados). Sólo un 5% de los encuestados considera que los modelos y estándares de calidad tienen influencia en la mencionada selección.
- Ninguno de los encuestados afirma tener conocimiento de modelos/estándares de calidad tales como: ISO 9126, IEEE 729, IEEE 610.12], ISO 14598, FURPS, Boehm, Dromey, McCall, Bansiya, ABAS, entre otros
- Relacionado a lo anterior, la mejor opción con la cual las personas encuestadas comprueban que el producto software alcanza un nivel adecuado de calidad, es la aceptación del cliente. Otra manera que consideran, con un 80% de preferencia para lograr lo anterior, es la Funcionalidad del producto. Ninguno de los encuestados considera que una certificación de calidad permite lograr el cometido en mención.
- Las fuentes consultadas o tenidas en cuenta para consolidar un conjunto de buenas prácticas en el desarrollo de productos software en las empresas son, en orden de importancia, Internet y tesis de grado, libros y guías de referencia y, por último, la experiencia del personal

ANEXO P: ENCUESTA DE CONVENIENCIA EN EL USO DE ATRIBUTOS DE CALIDAD PARA VALORAR EL PRODUCTO SOFTWARE

El presente anexo describe la encuesta realizada a un conjunto de MiPyMEs Desarrolladoras de Software de Popayán y Pasto. La razón de la creación de la encuesta obedece a la necesidad de conocer las opiniones de las MiPyMEs Desarrolladoras de Software en relación al conocimiento e interés de los atributos de calidad. Primero presenta el formato de la encuesta realizada y a continuación presentan los resultados obtenidos en cada una de las empresas. La encuesta fue realizada durante el mes de septiembre de 2009

P1. Diseño de la Encuesta

Encuesta: Conveniencia en el uso de atributos de calidad para valorar el producto software

Responsables:

MAG. Jorge Jair Moreno Chaustre

Estudiantes: Liliam Paola Bolaños Rengifo, Manuel Alejandro Navia Porras

Confidencialidad: Los responsables de esta encuesta se comprometen a no difundir la información personal o comercial a la que hayan accedido durante el desarrollo de la misma. Se difundirá de forma anónima los resultados y datos estadísticos generales sobre la información que arroje la encuesta.

Objetivo: Conocer acerca de los intereses en los atributos de calidad de las empresas. La siguiente encuesta pretende conocer cuáles son los atributos de calidad de software en los que las empresas del sur-occidente y proyectos universitarios dedicados al desarrollo de software, desean mayor enfoque.

DATOS DEL ENTREVISTADO	
Nombres Completos:	
Apellidos Completos:	
Correos Electrónicos:	
Organización:	
País:	
Ciudad:	
Fecha:	

1. De los atributos de calidad que se plantean a continuación seleccione la opción según su criterio en que es más importante que posean los productos software que desarrolla. Marque con una X en el recuadro según corresponda:

Nombre	Lo conoce		Puntaje											
	si	no	0	1	2	3	4	5	6	7	8	9	10	
Descripción														
Desempeño														
Evolución														
Fiabilidad (Confiabilidad)														

Nombre	Lo conoce		Puntaje											
	si	no	0	1	2	3	4	5	6	7	8	9	10	
Soportabilidad														
Usabilidad (Facilidad de uso)														
Corrección (Exactitud)														
Eficiencia														

Flexibilidad																				
Funcionalidad																				
Integridad (Seguridad Externa)																				
Portabilidad																				

Testeabilidad (Facilidad de prueba ó Chequeabilidad)																				
Interoperabilidad																				
Mantenibilidad (Facilidad de Mantenimiento)																				
Reusabilidad																				

2. Asociados a los atributos anteriores existen sub atributos. Seleccione solo aquellos que considere importantes. Marque con una X en el recuadro según corresponda:

Nombre	Lo conoce		Puntaje											
	si	no	0	1	2	3	4	5	6	7	8	9	10	
Abstracto (Abstracción)														
Accesibilidad														
Acceso de auditoría (Facilidad de auditoría)														
Acoplamiento														
Adaptabilidad														
Ajustable														
Amigabilidad														
Analizabilidad														
Asignado														
Atractivo														
Auto-Descripción														
Auto-Documentación														
Ayuda (Facilidad de ayuda)														
Capacidades														
Claridad														
Co existencia														
Cohesión														
Compatibilidad														
Complejidad														
Complejidad (Complejión)														
Comportamiento de Recursos														
Comportamiento en el tiempo														
Composición														
Computable														

Nombre	Lo conoce		Puntaje											
	si	no	0	1	2	3	4	5	6	7	8	9	10	
Expansibilidad (Extensibilidad)														
Explicitud														
Facilidad de aprendizaje														
Facilidad de Cambio														
Factores Humanos (Ingeniería humana)														
Formación (Facilidad de formación ó entrenamiento)														
Frecuencia y severidad de Fallos														
Generalidad														
Genérico *														
Gestionabilidad														
Herencia														
Homogéneo														
Idoneidad														
independencia														
Independencia de Hardware														
Independencia de Rango														
Independencia de Software														
Inicializado														
Instalabilidad														
Instrumentación														
Internalización (Localización)														
Jerarquía														
Legibilidad														
Madurez														

Nombre	Lo conoce		Puntaje											
	si	no	0	1	2	3	4	5	6	7	8	9	10	
Comunicatividad														
Concisión (conciso)														
Configurabilidad														
Conformidad														
Conjunto de Características														
Consistencia (CoHerencia)														
Control anomalías														
Control de acceso														
Crecimiento (Aumentabilidad)														
Degradabilidad														
Directo														
Diseño simple														
Disponibilidad														
Documentación														
Efectivo														
Eficiencia de Ejecución														
Eficiencia de almacenamiento														
Eficiencia de Dispositivo														
Encapsulamiento														
Comprensibilidad (Entendibilidad ó Facilidad de entendimiento)														
Especificado (Especificidad)														
Estabilidad														
Estandarización de las comunicaciones														
Estandarización de los datos														
Estética														
Estructuración														
Exactitud (Precisión)														

Nombre	Lo conoce		Puntaje											
	si	no	0	1	2	3	4	5	6	7	8	9	10	
Mensajes Electrónicos														
Modificabilidad														
Modularidad														
No redundante														
Operabilidad (operatividad)														
Parametrizado														
Personalización (Facilidad de personalizar)														
Polimorfismo														
Predictibilidad														
Progresivo														
Recuperabilidad														
Remplazabilidad														
Rendimiento														
Resuelto														
Seguridad (Seguridad interna)														
Servicio (Facilidad de servicio)														
Simplicidad														
Tamaño														
Tamaño de diseño														
Tiempo de recuperación														
Tiempo de respuesta														
Tiempo Medio entre Fallos														
Tolerancia al error (Tolerancia a fallos)														
Trazabilidad														
Uso de Recursos														
Utilizado														
Variante														
Velocidad														

3. Escriba sus comentarios y sugerencias (opcional)

Gracias por su atención y colaboración.

Firma del entrevistado

Firma del entrevistador

P2.Resultados de la Empresa SERATIC

DATOS DEL ENTREVISTADO	
Nombres Completos:	Diego Iván
Apellidos Completos:	Chamorro
Correos Electrónicos:	diego.chamorro@seratic.com
Organización:	SERATIC LTDA
País:	Colombia
Ciudad:	Popayán
Fecha:	15 septiembre de 2009

1. De los atributos de calidad que se plantean a continuación seleccione la opción según su criterio en que es más importante que posean los productos software que desarrolla. Marque con una X en el recuadro según corresponda:

Nombre	Lo conoce		Puntaje										
	si	no	0	1	2	3	4	5	6	7	8	9	10
Descripción	X												X
Desempeño	X												X
Evolución		X											
Fiabilidad (Confiabilidad)	X												X
Flexibilidad	X												X
Funcionalidad	X												X
Integridad (Seguridad Externa)	X							x					
Portabilidad	x												X

Nombre	Lo conoce		Puntaje										
	si	no	0	1	2	3	4	5	6	7	8	9	10
Soportabilidad		x											
Usabilidad (Facilidad de uso)	X												X
Corrección (Exactitud)		x											
Eficiencia	X											X	
Testeabilidad (Facilidad de prueba ó Chequeabilidad)		x		X									
Interoperabilidad	X											X	
Mantenibilidad (Facilidad de Mantenimiento)	X								X				
Reusabilidad	X											X	

2. Asociados a los atributos anteriores existen sub atributos. Seleccione solo aquellos que considere importantes. Marque con una X en el recuadro según corresponda:

Nombre	Lo conoce		Puntaje										
	si	no	0	1	2	3	4	5	6	7	8	9	10
Abstracto (Abstracción)		X											
Accesibilidad	X		x										
Acceso de auditoría (Facilidad de auditoría)		X											
Acoplamiento		X											
Adaptabilidad		X											

Nombre	Lo conoce		Puntaje										
	si	no	0	1	2	3	4	5	6	7	8	9	10
Expansibilidad (Extensibilidad)		X											
Explicitud		X											
Facilidad de aprendizaje	X											X	
Facilidad de Cambio		X											
Factores Humanos (Ingeniería humana)		X											

Nombre	Lo conoce		Puntaje										
	si	no	0	1	2	3	4	5	6	7	8	9	10
Ajustable		X											
Amigabilidad		X											
Analizabilidad	X											X	
Asignado		X											
Atractivo	X											X	
Auto-Descripción		X											
Auto-Documentación		X											
Ayuda (Facilidad de ayuda)	X			X									
Capacidades	X								X				
Claridad	X												
Co existencia	X								X				
Cohesión		X											
Compatibilidad	X			X									
Complejidad		X											
Complejidad (Complejión)	X			X									
Comportamiento de Recursos	X			X									
Comportamiento en el tiempo	X			X									
Composición		X											
Computable		X											
Comunicatividad	X												
Concisión (conciso)	X					X							
Configurabilidad	X								X				
Conformidad		X											
Conjunto de Características		X											
Consistencia (CoHerencia)	X					X							
Control anomalías	X					X							
Control de acceso	X						X						
Crecimiento (Aumentabilidad)		X											
Degradabilidad		X											
Directo		X											
Diseño simple	X											X	

Nombre	Lo conoce		Puntaje										
	si	no	0	1	2	3	4	5	6	7	8	9	10
Formación (Facilidad de formación ó entrenamiento)	X											X	
Frecuencia y severidad de Fallos	X								X				
Generalidad		X											
Genérico *		X											
Gestionabilidad	X											X	
Herencia	X										X		
Homogéneo		X											
Idoneidad		X											
Independencia		X											
Independencia de Hardware	X												X
Independencia de Rango	X												
Independencia de Software	X									X			
Inicializado		X											
Instalabilidad	X			X									
Instrumentación		X											
Internalización (Localización)	X								X				
Jerarquía		X											
Legibilidad		X											
Madurez		X											
Mensajes Electrónicos	X								X				
Modificabilidad		X											
Modularidad	X										X		
No redundante		X											
Operabilidad (operatividad)	X					X							
Parametrizado		X											
Personalización (Facilidad de personalizar)	X								X				
Polimorfismo	X											X	
Predictibilidad		X											
Progresivo	X			X									
Recuperabilidad	X					X							
Remplazabilidad	X			X									

Nombre	Lo conoce		Puntaje										
	si	no	0	1	2	3	4	5	6	7	8	9	10
Disponibilidad	X				X								
Documentación	X					X							
Efectivo	X					X							
Eficiencia de Ejecución	X						X						
Eficiencia de almacenamiento	X									X			
Eficiencia de Dispositivo	X							X					
Encapsulamiento	X					X							
Comprensibilidad (Entendibilidad ó Facilidad de entendimiento)		X											
Especificado (Especificidad)		X											
Estabilidad		X											
Estandarización de las comunicaciones	X							X					
Estandarización de los datos	X					X							
Estética	X											X	
Estructuración	X								X				
Exactitud (Precisión)	x		X										

Nombre	Lo conoce		Puntaje										
	si	no	0	1	2	3	4	5	6	7	8	9	10
Rendimiento	X									X			
Resuelto		X											
Seguridad (Seguridad interna)	X						X						
Servicio (Facilidad de servicio)	X		X										
Simplicidad	X			X									
Tamaño	X		X										
Tamaño de diseño	X		X										
Tiempo de recuperación	X				X								
Tiempo de respuesta	X							X					
Tiempo Medio entre Fallos		X											
Tolerancia al error (Tolerancia a fallos)	X				X								
Trazabilidad	X				X								
Uso de Recursos	X												X
Utilizado		X											
Variante	X		X										
Velocidad	X												x

3. Escriba sus comentarios y sugerencias (opcional). _____

P3.Resultados de la Empresa INPUT TECHNOLOGIES

DATOS DEL ENTREVISTADO	
Nombres Completos:	RICARDO ANDRÉS
Apellidos Completos:	LEDEZMA CHAVEZ
Correos Electrónicos:	ricardo.ledezma@input.com.co
Organización:	INPUT TECHNOLOGIES LTDA.
País:	COLOMBIA
Ciudad:	POPAYÁN
Fecha:	10 septiembre de 2009

- De los atributos de calidad que se plantean a continuación seleccione la opción según su criterio en que es más importante que posean los productos software que desarrolla. Marque con una X en el recuadro según corresponda:

Nombre	Lo conoce		Puntaje											
	si	no	0	1	2	3	4	5	6	7	8	9	10	
Descripción		X												
Desempeño		X												
Evolución		X												
Fiabilidad (Confiabilidad)		X												
Flexibilidad		X												
Funcionalidad	X											X		
Integridad (Seguridad Externa)	X											X		
Portabilidad		x												

Nombre	Lo conoce		Puntaje											
	si	no	0	1	2	3	4	5	6	7	8	9	10	
Soportabilidad		X												
Usabilidad (Facilidad de uso)	X												X	
Corrección (Exactitud)		X												
Eficiencia		X												
Testeabilidad (Facilidad de prueba ó Chequeabilidad)		X												
Interoperabilidad	X											X		
Mantenibilidad (Facilidad de Mantenimiento)	X											X		
Reusabilidad		X												

2. Asociados a los atributos anteriores existen sub atributos. Seleccione solo aquellos que considere importantes. Marque con una X en el recuadro según corresponda:

Nombre	Lo conoce		Puntaje											
	si	no	0	1	2	3	4	5	6	7	8	9	10	
Abstracto (Abstracción)		X												
Accesibilidad		X												
Acceso de auditoría (Facilidad de auditoría)		X												
Acoplamiento		X												
Adaptabilidad		X												
Ajustable	X										X			
Amigabilidad		X												
Analizabilidad		X												
Asignado		X												
Atractivo	X										X			
Auto-Descripción		X												
Auto-Documentación		X												
Ayuda (Facilidad de ayuda)	X											X		
Capacidades		X												
Claridad	X											X		
Co existencia		X												
Cohesión		X												

Nombre	Lo conoce		Puntaje											
	si	no	0	1	2	3	4	5	6	7	8	9	10	
Expansibilidad (Extensibilidad)		X												
Explicitud	X											X		
Facilidad de aprendizaje	X											X		
Facilidad de Cambio		X												
Factores Humanos (Ingeniería humana)		X												
Formación (Facilidad de formación ó entrenamiento)	X													
Frecuencia y severidad de Fallos		X												
Generalidad		X												
Genérico *		X												
Gestionabilidad		X												
Herencia		X												
Homogéneo		X												
Idoneidad		X												
independencia		X												
Independencia de Hardware	X													
Independencia de Rango		X												
Independencia de Software		X												

Nombre	Lo conoce		Puntaje										
	si	no	0	1	2	3	4	5	6	7	8	9	10
Compatibilidad		X											
Complejidad		X											
Complejidad (Complejión)		X											
Comportamiento de Recursos		X											
Comportamiento en el tiempo		X											
Composición		X											
Computable		X											
Comunicatividad		X											
Concisión (conciso)		X											
Configurabilidad		X											
Conformidad		X											
Conjunto de Características		X											
Consistencia (CoHerencia)		X											
Control anomalías		X											
Control de acceso	X											X	
Crecimiento (Aumentabilidad)		X											
Degradabilidad		X											
Directo		X											
Diseño simple		X											
Disponibilidad		X											
Documentación		X											
Efectivo		X											
Eficiencia de Ejecución	X										X		
Eficiencia de almacenamiento		X											
Eficiencia de Dispositivo		X											
Encapsulamiento		X											
Comprensibilidad (Entendibilidad ó Facilidad de entendimiento)	X										X		
Especificado (Especificidad)		X											
Estabilidad		X											

Nombre	Lo conoce		Puntaje										
	si	no	0	1	2	3	4	5	6	7	8	9	10
Inicializado		X											
Instalabilidad		X											
Instrumentación		X											
Internalización (Localización)		X											
Jerarquía		X											
Legibilidad		X											
Madurez		X											
Mensajes Electrónicos		X											
Modificabilidad		X											
Modularidad	X										X		
No redundante		X											
Operabilidad (operatividad)		X											
Parametrizado		X											
Personalización (Facilidad de personalizar)		X											
Polimorfismo		X											
Predictibilidad		X											
Progresivo		X											
Recuperabilidad		X											
Remplazabilidad		X											
Rendimiento		X											
Resuelto		X											
Seguridad (Seguridad interna)		X											
Servicio (Facilidad de servicio)	X											X	
Simplicidad		X											
Tamaño		X											
Tamaño de diseño		X											
Tiempo de recuperación	X											X	
Tiempo de respuesta	X												X
Tiempo Medio entre Fallos	X												X

Nombre	Lo conoce		Puntaje											
	si	no	0	1	2	3	4	5	6	7	8	9	10	
Estandarización de las comunicaciones		X												
Estandarización de los datos	X											X		
Estética	X										X			
Estructuración		X												
Exactitud (Precisión)		x												

Nombre	Lo conoce		Puntaje											
	si	no	0	1	2	3	4	5	6	7	8	9	10	
Tolerancia al error (Tolerancia a fallos)	X												X	
Trazabilidad		X												
Uso de Recursos		X												
Utilizado		X												
Variante		X												
Velocidad		x												

3. Escriba sus comentarios y sugerencias (opcional). _____

P4.Resultados de la Empresa LATIN BUSINESS

DATOS DEL ENTREVISTADO	
Nombres Completos:	John Jairo
Apellidos Completos:	Ortiz Canchala
Correos Electrónicos:	jjortiz@parquesoftpasto.com, jjortiz1@gmail.com
Organización:	Latin Business Ltda
País:	Colombia
Ciudad:	Pasto
Fecha:	18 – 09 - 2009

1. De los atributos de calidad que se plantean a continuación seleccione la opción según su criterio en que es más importante que posean los productos software que desarrolla. Marque con una X en el recuadro según corresponda:

Nombre	Lo conoce		Puntaje											
	si	no	0	1	2	3	4	5	6	7	8	9	10	
Descripción	X							X						
Desempeño	X									X				
Evolución	X							X						
Fiabilidad (Confiabilidad)	X										X			
Flexibilidad	X										X			
Funcionalidad	X										X			
Integridad (Seguridad Externa)	X										X			
Portabilidad	X											X		

Nombre	Lo conoce		Puntaje											
	si	no	0	1	2	3	4	5	6	7	8	9	10	
Soportabilidad		X												
Usabilidad (Facilidad de uso)	X												X	
Corrección (Exactitud)	X												X	
Eficiencia	X											X		
Testeabilidad (Facilidad de prueba ó Chequeabilidad)	X											X		
Interoperabilidad	X											X		
Mantenibilidad (Facilidad de Mantenimiento)	X											X		
Reusabilidad	X											X		

2. Asociados a los atributos anteriores existen sub atributos. Seleccione solo aquellos que considere importantes. Marque con una X en el recuadro según corresponda:

Nombre	Lo conoce		Puntaje										
	si	no	0	1	2	3	4	5	6	7	8	9	10
Abstracto (Abstracción)	X							X					
Accesibilidad	X								X				
Acceso de auditoría (Facilidad de auditoría)		X											
Acoplamiento		X											
Adaptabilidad	X										X		
Ajustable	X									X			
Amigabilidad	X									X			
Analizabilidad		X											
Asignado		X											
Atractivo	X									X			
Auto-Descripción		X											
Auto-Documentación		X											
Ayuda (Facilidad de ayuda)	X							X					
Capacidades		X											
Claridad		X											
Co existencia		X											
Cohesión		X											
Compatibilidad	X								X				
Complejidad	X								X				
Complejidad (Complejión)		X											
Comportamiento de Recursos		X											
Comportamiento en el tiempo		X											
Composición		X											
Computable		X											
Comunicatividad		X											
Concisión (conciso)		X											
Configurabilidad	X									X			
Conformidad	X									X			
Conjunto de Características	X									X			

Nombre	Lo conoce		Puntaje										
	si	no	0	1	2	3	4	5	6	7	8	9	10
Expansibilidad (Extensibilidad)	X										X		
Explicitud		X											
Facilidad de aprendizaje	X											X	
Facilidad de Cambio	X										X		
Factores Humanos (Ingeniería humana)	X										X		
Formación (Facilidad de formación ó entrenamiento)	X												X
Frecuencia y severidad de Fallos	X										X		
Generalidad		X											
Genérico *		X											
Gestionabilidad	X										X		
Herencia	X							X					
Homogéneo		X											
Idoneidad		X											
independencia	X									X			
Independencia de Hardware	X												X
Independencia de Rango		X											
Independencia de Software		X											
Inicializado		X											
Instalabilidad	X											X	
Instrumentación		X											
Internalización (Localización)	X									X			
Jerarquía		X											
Legibilidad		X											
Madurez		X											
Mensajes Electrónicos		X											
Modificabilidad	X										X		
Modularidad	X										X		
No redundante	X										X		
Operabilidad (operatividad)	X											X	

Nombre	Lo conoce		Puntaje										
	si	no	0	1	2	3	4	5	6	7	8	9	10
Consistencia (CoHerencia)	X											X	
Control anomalías	X											X	
Control de acceso													
Crecimiento (Aumentabilidad)	X											X	
Degradabilidad	X											X	
Directo		X											
Diseño simple	X											X	
Disponibilidad	X											X	
Documentación	X												X
Efectivo		X											
Eficiencia de Ejecución	X											X	
Eficiencia de almacenamiento	X											X	
Eficiencia de Dispositivo	X											X	
Encapsulamiento		X											
Comprensibilidad (Entendibilidad ó Facilidad de entendimiento)	X											X	
Especificado (Especificidad)	X											X	
Estabilidad	X												X
Estandarización de las comunicaciones	X											X	
Estandarización de los datos	X											X	
Estética	X											X	
Estructuración	X											X	
Exactitud (Precisión)	X												X

Nombre	Lo conoce		Puntaje										
	si	no	0	1	2	3	4	5	6	7	8	9	10
Parametrizado	X												X
Personalización (Facilidad de personalizar)	X												X
Polimorfismo													
Predictibilidad	X											X	
Progresivo		X											
Recuperabilidad	X										X		
Remplazabilidad	X										X		
Rendimiento	X										X		
Resuelto		X											
Seguridad (Seguridad interna)	X											X	
Servicio (Facilidad de servicio)	X												X
Simplicidad	X												X
Tamaño	X												X
Tamaño de diseño	X												X
Tiempo de recuperación	X												X
Tiempo de respuesta	X												X
Tiempo Medio entre Fallos	X												X
Tolerancia al error (Tolerancia a fallos)	X												X
Trazabilidad		X											
Uso de Recursos	X											X	
Utilizado		X											
Variante		X											
Velocidad	X												X

Escriba sus comentarios y sugerencias (opcional). Algunos atributos no los reconocí por su nombre por tal motivo los marqué como "no lo conoce", y alguno otros no tenía muy claro si la idea o percepción que tengo de ellos es la correcta, por lo cual los marque de la misma manera, ojala y en la visita que tienen

P5.Resultados de la Empresa SITI

DATOS DEL ENTREVISTADO

Nombres Completos:	Gelber
Apellidos Completos:	Moran Silva
Correos Electrónicos:	gebermoran@siticol.com
Organización:	SITI - Soluciones Integrales en Tecnologías de la información
País:	Colombia
Ciudad:	Pasto
Fecha:	20 septiembre de 2009

4. De los atributos de calidad que se plantean a continuación seleccione la opción según su criterio en que es más importante que posean los productos software que desarrolla. Marque con una X en el recuadro según corresponda:

Nombre	Lo conoce		Puntaje											
	si	no	0	1	2	3	4	5	6	7	8	9	10	
Descripción													X	
Desempeño														X
Evolución														
Fiabilidad (Confiabilidad)													X	
Flexibilidad													X	
Funcionalidad														X
Integridad (Seguridad Externa)							X							
Portabilidad													X	

Nombre	Lo conoce		Puntaje											
	si	no	0	1	2	3	4	5	6	7	8	9	10	
Soportabilidad														
Usabilidad (Facilidad de uso)														X
Corrección (Exactitud)														
Eficiencia												X		
Testeabilidad (Facilidad de prueba ó Chequeabilidad)								X						
Interoperabilidad											X			
Mantenibilidad (Facilidad de Mantenimiento)									X					
Reusabilidad											X			

5. Asociados a los atributos anteriores existen sub atributos. Seleccione solo aquellos que considere importantes. Marque con una X en el recuadro según corresponda:

Nombre	Lo conoce		Puntaje											
	si	no	0	1	2	3	4	5	6	7	8	9	10	
Abstracto (Abstracción)													X	
Accesibilidad													X	
Acceso de auditoría (Facilidad de auditoría)								X						
Acoplamiento									X					
Adaptabilidad									X					
Ajustable												X		
Amigabilidad													X	
Analizabilidad												X		
Asignado									X					
Atractivo													X	
Auto-Descripción				X										

Nombre	Lo conoce		Puntaje											
	si	no	0	1	2	3	4	5	6	7	8	9	10	
Expansibilidad (Extensibilidad)												X		
Explicitud												X		
Facilidad de aprendizaje													X	
Facilidad de Cambio													X	
Factores Humanos (Ingeniería humana)													X	
Formación (Facilidad de formación ó entrenamiento)												X		
Frecuencia y severidad de Fallos												X		
Generalidad												X		
Genérico *												X		
Gestionabilidad													X	
Herencia														

Nombre	Lo conoce		Puntaje										
	si	no	0	1	2	3	4	5	6	7	8	9	10
Auto-Documentación				X									
Ayuda (Facilidad de ayuda)									X				
Capacidades											X		
Claridad												X	
Co existencia										X			
Cohesión													
Compatibilidad											X		
Complejidad									X				
Complejidad (Complejión)													X
Comportamiento de Recursos											X		
Comportamiento en el tiempo											X		
Composición													
Computable													
Comunicatividad													
Concisión (conciso)												X	
Configurabilidad										X			
Conformidad												X	
Conjunto de Características													
Consistencia (CoHerencia)												X	
Control anomalías													X
Control de acceso													
Crecimiento (Aumentabilidad)									X				
Degradabilidad										X			
Directo													
Diseño simple												X	
Disponibilidad												X	
Documentación										X			
Efectivo													X
Eficiencia de Ejecución											X		
Eficiencia de almacenamiento										X			
Eficiencia de Dispositivo											X		

Nombre	Lo conoce		Puntaje										
	si	no	0	1	2	3	4	5	6	7	8	9	10
Homogéneo												X	
Idoneidad												X	
Independencia												X	
Independencia de Hardware													X
Independencia de Rango												X	
Independencia de Software											X		
Inicializado													
Instalabilidad										X			
Instrumentación													
Internalización (Localización)												X	
Jerarquía													
Legibilidad											X		
Madurez													
Mensajes Electrónicos									X				
Modificabilidad												X	
Modularidad									X				
No redundante												X	
Operabilidad (operatividad)												X	
Parametrizado												X	
Personalización (Facilidad de personalizar)												X	
Polimorfismo											X		
Predictibilidad										X			
Progresivo										X			
Recuperabilidad										X			
Remplazabilidad													
Rendimiento													X
Resuelto													X
Seguridad (Seguridad interna)										X			
Servicio (Facilidad de servicio)												X	
Simplicidad												X	
Tamaño												X	

Nombre	Lo conoce		Puntaje										
	si	no	0	1	2	3	4	5	6	7	8	9	10
Encapsulamiento											x		
Comprensibilidad (Entendibilidad ó Facilidad de entendimiento)												X	
Especificado (Especificidad)											x		
Estabilidad												X	
Estandarización de las comunicaciones										X			
Estandarización de los datos											X		
Estética												X	
Estructuración												X	
Exactitud (Precisión)													

Nombre	Lo conoce		Puntaje										
	si	no	0	1	2	3	4	5	6	7	8	9	10
Tamaño de diseño										X			
Tiempo de recuperación													X
Tiempo de respuesta													X
Tiempo Medio entre Fallos										X			
Tolerancia al error (Tolerancia a fallos)											X		
Trazabilidad											X		
Uso de Recursos											X		
Utilizado													
Variante													
Velocidad												X	

6. Escriba sus comentarios y sugerencias (opcional).

P6.Resultados generales de las empresas

	Nombre	Empresa	Fecha de encuesta
Encuestado 1	Diego Chamorro	SERATIC	15-Sep-09
Encuestado 2	Gelber Moran	SITI	20-Sep-09
Encuestado 3	Jhon Jairo Ortiz	LATIN BUSINESS	18-Sep-09
Encuestado 4	Ricardo Ledezma	INPUT	10-Sep-09

Nombre	Encuestado 1			Encuestado 1			Encuestado 1			Encuestado 1			Prom.
	Lo conoce		Punt	Lo conoce		Punt	Lo conoce		Punt	Lo conoce		Punt	
	si	no		si	no		si	no		si	no		
Descripción	x		9	x		9	x		5		x		7,7
Desempeño	x		10	x		9	x		7		x		8,7
Evolución		x		x		8	x		5		x		6,5
Fiabilidad (Confiabilidad)	x		9	x		9	x		8		x		8,7
Flexibilidad	x		9	x		9	x		8		x		8,7
Funcionalidad	x		10	x		10	x		8	x		8	9,0
Integridad (Seguridad Externa)	x		5	x		10	x		8	x		8	7,8
Portabilidad	x		9	x		8	x		9		x		8,7
Soportabilidad		x		x		8		x			x		8,0
Usabilidad (Facilidad de uso)	x		10	x		10	x		9	x		9	9,5
Corrección (Exactitud)		x		x		7	x		9		x		8,0
Eficiencia	x		7	x		9	x		8		x		8,0

Testeabilidad (Facilidad de prueba ó Chequeabilidad)	x		1	x		9	x		8		x		6,0
Interoperabilidad	x		7	x		9	x		8	x		7	7,8
Mantenibilidad (Facilidad de Mantenimiento)	x		5	x		8	x		7	x		7	6,8
Reusabilidad	x		7	x		8	x		7		x		7,3

Tabla 35. Promedio por Atributo

Nombre	Encuestado 1		Encuestado 2		Encuestado 3		Encuestado 4		Promedio				
	Lo conoce		Lo conoce		Lo conoce		Lo conoce						
	si	no	Puntaje	si	no	Puntaje	si	no		Puntaje			
Abstracto (Abstracción)		x		x		9	x		5		x		7,0
Accesibilidad	x		0	x		9	x		6		x		5,0
Acceso de auditoría (Facilidad de auditoría)		x		x		5		x			x		5,0
Acoplamiento		x		x		6		x			x		6,0
Adaptabilidad		x		x		6	x		8		x		7,0
Ajustable		x		x		8	x		8	x		7	7,7
Amigabilidad	x		9	x		9	x		8		x		8,7
Analizabilidad		x		x		8		x			x		8,0
Asignado		x		x		6		x			x		6,0
Atractivo	x		9	x		9	x		8	x		8	8,5
Auto-Descripción		x		x		2		x			x		2,0
Auto-Documentación		x		x		1		x			x		1,0
Ayuda (Facilidad de ayuda)	x		2	x		6	x		6	x		8	5,5
Capacidades	x		7	x		8		x			x		7,5
Claridad		x		x		9		x		x		8	8,5
Co existencia	x		7	x		7		x			x		7,0
Cohesión		x			x			x			x		
Compatibilidad	x		1	x		8	x		7		x		5,3
Complejidad		x		x		6	x		7		x		6,5
Compleitud (Compleción)		x		x		10		x			x		10,0
Comportamiento de Recursos	x		1	x		8		x			x		4,5
Comportamiento en el tiempo	x		2	x		8		x			x		5,0
Composición		x			x			x			x		
Computable		x						x			x		
Comunicatividad		x			x			x			x		
Concisión (conciso)	x		4	x		9		x			x		6,5
Configurabilidad	x		7	x		7	x		8		x		7,3
Conformidad		x		x		10	x		8		x		9,0

Nombre	Encuestado 1			Encuestado 2			Encuestado 3			Encuestado 4			Promedio
	Lo conoce		Puntaje	Lo conoce		Puntaje	Lo conoce		Puntaje	Lo conoce		Puntaje	
	si	no		si	no		si	no		si	no		
Conjunto de Características		x			x		x			x		8,0	
Consistencia (CoHerencia)	x		4	x		9	x		8	x		7,0	
Control anomalías	x		4	x		10	x		8	x		7,3	
Control de acceso	x		5		x			x		x		9	7,0
Crecimiento (Aumentabilidad)		x		x		7	x		7	x		7,0	
Degradabilidad		x		x		7	x		7	x		7,0	
Directo					x			x		x			
Diseño simple	x		9	x		9	x		8	x		8,7	
Disponibilidad	x		2	x		9	x		8	x		6,3	
Documentación	x		6	x		7	x		9	x		7,3	
Efectivo	x		4	x		10		x		x		7,0	
Eficiencia de Ejecución	x		4	x		8	x		9	x		8	7,3
Eficiencia de almacenamiento	x		5	x		7	x		9	x		7,0	
Eficiencia de Dispositivo	x		8	x		8	x		7	x		7,7	
Encapsulamiento	x		6	x		7		x		x		6,5	
Comprensibilidad (Entendibilidad ó Facilidad de entendimiento)	x		4	x		8	x		8	x		8	7,0
Especificado (Especificidad)		x		x		8	x		7	x		7,5	
Estabilidad		x		x		8	x		10	x		9,0	
Estandarización de las comunicaciones	x		6	x		6	x		7	x		6,3	
Estandarización de los datos	x		4	x		7	x		7	x		8	6,5
Estética	x		9	x		9	x		7	x		7	8,0
Estructuración	x		7	x		8	x		8	x		7,7	
Exactitud (Precisión)	x		0	x			x		9	x		4,5	
Expansibilidad (Extensibilidad)		x		x		7	x		8	x		7,5	
Explicitud		x			x			x		x			
Facilidad de aprendizaje	x		7	x		8	x		9	x		8	8,0
Facilidad de Cambio		x		x		9	x		8	x		8	8,3
Factores Humanos (Ingeniería humana)		x		x		9	x		8	x			8,5
Formación (Facilidad de formación ó entrenamiento)	x		7	x		8	x		10	x			8,3
Frecuencia y severidad de Fallos	x		5	x		8	x		8	x			7,0

Nombre	Encuestado 1			Encuestado 2			Encuestado 3			Encuestado 4			Promedio
	Lo conoce		Puntaje	Lo conoce		Puntaje	Lo conoce		Puntaje	Lo conoce		Puntaje	
	si	no		si	no		si	no		si	no		
Generalidad		x		x		8		x		x		8,0	
Genérico *		x		x		8				x		8,0	
Gestionabilidad	x		8		x		x		8		x	8,0	
Herencia	x		7		x		x		6		x	6,5	
Homogéneo		x		x		7		x		x		7,0	
Idoneidad		x		x		7		x		x		7,0	
independencia		x		x		8	x		7		x	7,5	
Independencia de Hardware	x		9	x		9	x		10	x	0	7,0	
Independencia de Rango		x		x		8		x		x		8,0	
Independencia de Software	x		6	x		6		x		x		6,0	
Inicializado		x			x			x		x			
Instabilidad	x		1	x		5	x		9		x	5,0	
Instrumentación		x			x			x		x			
Internalización (Localización)	x		5	x		7	x		7		x	6,3	
Jerarquía		x			x			x		x			
Legibilidad		x		x		6		x		x		6,0	
Madurez		x			x			x		x			
Mensajes Electrónicos	x		5	x		3		x		x		4,0	
Modificabilidad		x		x		8	x		8		x	8,0	
Modularidad	x		7	x		1	x		8	x	7	5,8	
No redundante	x		2	x		7	x		8		x	5,7	
Operabilidad (operatividad)		x		x		8	x		9		x	8,5	
Parametrizado	x		5	x		8	x		9		x	7,3	
Personalización (Facilidad de personalizar)	x		8	x		8	x		9		x	8,3	
Polimorfismo		x		x		7		x		x		7,0	
Predictibilidad		x		x		4	x		8		x	6,0	
Progresivo	x		0	x		4		x		x		2,0	
Recuperabilidad	x		2	x		5	x		6		x	4,3	
Reemplazabilidad	x		0		x		x		7		x	3,5	
Rendimiento	x		6	x		9	x		7		x	7,3	
Resuelto		x		x		9		x		x		9,0	
Seguridad (Seguridad interna)	x		3	x		1	x		8		x	4,0	
Servicio (Facilidad de servicio)	x		0	x		8	x		9	x	8	6,3	
Simplicidad	x		1	x		8	x		9		x	6,0	

Nombre	Encuestado 1		Puntaje	Encuestado 2		Puntaje	Encuestado 3		Puntaje	Encuestado 4		Puntaje	Promedio
	Lo conoce			Lo conoce			Lo conoce			Lo conoce			
	si	no	si	no	si	no	si	no					
Tamaño	x		0	x		7	x		9		x		5,3
Tamaño de diseño	x		0	x		2	x		9		x		3,7
Tiempo de recuperación	x		2	x		8	x		9	x		8	6,8
Tiempo de respuesta	x		5	x		8	x		9	x		9	7,8
Tiempo Medio entre Fallos		x		x		1	x		9	x		9	6,3
Tolerancia al error (Tolerancia a fallos)	x		1	x		5	x		9	x		8	5,8
Trazabilidad	x		1	x		4		x			x		2,5
Uso de Recursos	x		10	x		4	x		7		x		7,0
Utilizado		x			x			x			x		
Variante	x		0		x			x			x		0,0
Velocidad	x		9	x		6	x		8		x		7,7

Tabla 36. Promedio por Sub-Atributo

ANEXO Q: APLICACIÓN DEL MARCO EN LAS EMPRESAS

El presente anexo describe la selección de atributos y métricas a aplicar en un subproducto generado por las empresas, esto según lo visto en el anexo Q (encuesta de atributo), y los resultados arrojados en el momento de aplicar las métricas. Las visitas fueron realizadas en los meses de octubre y noviembre de 2009.

Q1. Selección de atributos y métricas.

Plantilla: Atributo, sub atributos y métricas						
Atributo	Sub-atributo	Métrica	V. Mín.	V. Máx.	V. Evaluación	Recomendación
Confiabilidad	Encapsulamiento	Métrica de acceso de datos (DAM)	0	1		
		Method hiding factor- MHF	0	1		
		Attribute hiding factor- AHF	0	1		
	Acoplamiento	Coupling Between objects (CBO)				
Coupling Factor - COF		0	1			
Mantenibilidad	Encapsulamiento	Métrica de acceso de datos (DAM)	0	1		
		Method hiding factor- MHF	0	1		
		Attribute hiding factor- AHF	0	1		
	Reusabilidad	Coupling Between objects (CBO)				
		Number of children NOC				
Funcionalidad	Seguridad	Access controllability	0	1		
		Data encryption	0	1		
	Tamaño	Size2				
		Numero de clases				
Usabilidad	Auto descripción	Polymorphism Factor -POF	0	1		
		Medidas de Boehm para auto descripción				
	Facilidad de aprendizaje	Quantity of comments (Por modulo)	0	1		
		Effectiveness of the user documentation and/or help system	0	1		
		Effectiveness of the user documentation and/or help system in use	0	1		

Q2. Resultados de la Empresa SERATIC

Plantilla: Atributo, sub atributos y métricas						
Atributo	Sub-atributo	Métrica	V. Mín.	V. Máx.	V. Evaluación	Recomendación
Confiabilidad	Encapsulamiento	Métrica de acceso de datos (DAM)	0	1	0,7	
		Method hiding factor- MHF	0	1	0,3	
		Attribute hiding factor- AHF	0	1	0,2	
	Acoplamiento	Coupling Between objects (CBO)			18,5	
Coupling Factor - COF		0	1	0,05		
Mantenibilidad	Encapsulamiento	Métrica de acceso de datos (DAM)	0	1	0,7	
		Method hiding factor- MHF	0	1	0,3	
		Attribute hiding factor- AHF	0	1	0,2	
	Reusabilidad	Coupling Between objects			18,5	

		(CBO)				
		Number of children NOC			3,2	
Funcionalidad	Seguridad	Access controllability	0	1	1	
		Data encryption	0	1	1	
	Tamaño	Size2			46,9	
		Numero de clases			11	
	Polimorfismo	Polymorphism Factor -POF	0	1	NA	
Usabilidad	Auto descripción	Medidas de Boehm para auto descripción			0,3	
		Quantity of comments (Por modulo)	0	1	NA	
	Facilidad de aprendizaje	Effectiveness of the user documentation and/or help system	0	1	NA	
		Effectiveness of the user documentation and/or help system in use	0	1	NA	

Q3. Resultados de la Empresa INPUT TECHNOLOGIES

Plantilla: Atributo, sub atributos y métricas						
Atributo	Sub-atributo	Métrica	V. Mín	V. Máx	V. Evaluación	Recomendación
Confiabilidad	Encapsulamiento	Métrica de acceso de datos (DAM)	0	1	0,3	
		Method hiding factor- MHF	0	1	NA	
		Attribute hiding factor- AHF	0	1	0,6	
	Acoplamiento	Coupling Between objects (CBO)			2,5	
Coupling Factor - COF		0	1	0,1		
Mantenibilidad	Encapsulamiento	Métrica de acceso de datos (DAM)	0	1	0,3	
		Method hiding factor- MHF	0	1	NA	
		Attribute hiding factor- AHF	0	1	0,6	
	Reusabilidad	Coupling Between objects (CBO)			2,5	
		Number of children NOC			0	
Funcionalidad	Seguridad	Access controllability	0	1	1	
		Data encryption	0	1	NA	
	Tamaño	Size2			4,4	
		Numero de clases			8	
		Polimorfismo	Polymorphism Factor -POF	0	1	NA
Usabilidad	Auto descripción	Medidas de Boehm para auto descripción			0,2	
		Quantity of comments (Por modulo)	0	1	NA	
	Facilidad de aprendizaje	Effectiveness of the user documentation and/or help system	0	1	NA	
		Effectiveness of the user documentation and/or help system in use	0	1	NA	

Q4. Resultados de la Empresa LATIN BUSINESS

Plantilla: Atributo, sub atributos y métricas						
Atributo	Sub-atributo	Métrica	V. Mín	V. Máx	V. Evaluación	Recomendación
Confiabilidad	Encapsulamiento	Métrica de acceso de datos (DAM)	0	1	0,3	
		Method hiding factor- MHF	0	1	0,1	

	Acoplamiento	Attribute hiding factor- AHF	0	1	0,04	
		Coupling Between objects (CBO)			4,1	
		Coupling Factor - COF	0	1	0,3	
Mantenibilidad	Encapsulamiento	Métrica de acceso de datos (DAM)	0	1	0,3	
		Method hiding factor- MHF	0	1	0,1	
		Attribute hiding factor- AHF	0	1	0,04	
	Reusabilidad	Coupling Between objects (CBO)			4,1	
		Number of children NOC			3	
Funcionalidad	Seguridad	Access controllability	0	1	0,7	
		Data encryption	0	1	1	
	Tamaño	Size2			7,5	
		Numero de clases			7	
	Polimorfismo	Polymorphism Factor -POF	0	1	0,02	
Usabilidad	Auto descripción	Medidas de Boehm para auto descripción			0,5	
		Quantity of comments (Por modulo)	0	1	NA	
	Facilidad de aprendizaje	Effectiveness of the user documentation and/or help system	0	1	NA	
		Effectiveness of the user documentation and/or help system in use	0	1	0,7	

Q5. Resultados de la Empresa SITI

Plantilla: Atributo, sub atributos y métricas						
Atributo	Sub-atributo	Métrica	V. Mín	V. Máx	V. Evaluación	Recomendación
Confiabilidad	Encapsulamiento	Métrica de acceso de datos (DAM)	0	1	1	
		Method hiding factor- MHF	0	1	0,06	
		Attribute hiding factor- AHF	0	1	0,4	
	Acoplamiento	Coupling Between objects (CBO)			2,1	
		Coupling Factor - COF	0	1	0,09	
Mantenibilidad	Encapsulamiento	Métrica de acceso de datos (DAM)	0	1	1	
		Method hiding factor- MHF	0	1	0,06	
		Attribute hiding factor- AHF	0	1	0,4	
	Reusabilidad	Coupling Between objects (CBO)			2,1	
		Number of children NOC			1,2	
Funcionalidad	Seguridad	Access controllability	0	1	1	
		Data encryption	0	1	1	
	Tamaño	Size2			25,2	
		Numero de clases			12	
	Polimorfismo	Polymorphism Factor -POF	0	1	0	
Usabilidad	Auto descripción	Medidas de Boehm para auto descripción			0,625	
		Quantity of comments (Por modulo)	0	1	0,11	
	Facilidad de aprendizaje	Effectiveness of the user documentation and/or help system	0	1	0	
		Effectiveness of the user documentation and/or help system in use	0	1	0,85	

Q6. Resultados generales de las empresas

A continuación es presentado a manera de informe los resultados generales obtenidos por empresa, y las observaciones que pueden obtenerse de dichos resultados. Los artefactos suministrados por cada empresa son:

- Organización: INPUT Technologies Ltda.– Popayán
Representante: Ricardo Andres Ledezma
Producto: Compromiso
Modulo: Actas
- Organización: Latin Business Ltda – Pasto.
Representante: Jhon Jairo Ortiz
Producto: Truck Dispatch
Modulo: Controladores
- Organización: SERATIC Ltda.– Popayán
Representante: Diego Chamorro
Producto: Easy Sales
Modulo: Módulo de Pedidos
- Organización: SITI Ltda - Pasto
Representante: Gelber Moran
Producto: SIGE – Sistema de Información para la Gestión empresarial Modulo:
Centro de Documentación
Componente: Correspondencia de documentos de entrada

En la tabla siguiente pueden verse los resultados por empresa.

Plantilla: Atributo, sub atributos y métricas									
Atributo	Sub-atributo	Métrica	V. Mín	V. Máx	SERATIC	INPUT	LATIN	SITI	Prom
Confiabilidad	Encapsulamiento	Métrica de acceso de datos (DAM)	0	1	0,7	0,3	0,3	1	0,58
		Method hiding factor- MHF	0	1	0,3	NA	0,1	0,06	0,15
		Attribute hiding factor- AHF	0	1	0,2	0,6	0,04	0,4	0,31
	Acoplamiento	Coupling Between objects (CBO)			18,5	2,5	4,1	2,1	6,80
		Coupling Factor - COF	0	1	0,05	0,1	0,3	0,09	0,14
Mantenibilidad	Encapsulamiento	Métrica de acceso de datos (DAM)	0	1	0,7	0,3	0,3	1	0,58
		Method hiding factor- MHF	0	1	0,3	NA	0,1	0,06	0,15
		Attribute hiding factor- AHF	0	1	0,2	0,6	0,04	0,4	0,31
	Reusabilidad	Coupling Between objects (CBO)			18,5	2,5	4,1	2,1	6,80
		Number of children NOC			3,2	0	3	1,2	1,85
Funcionalidad	Seguridad	Access controllability	0	1	1	1	0,7	1	0,93

		Data encryption	0	1	1	NA	1	1	1,00
	Tamaño	Size2			46,9	4,4	7,5	25,2	21,00
		Numero de clases			11	8	7	12	9,50
	Polimorfismo	Polymorphism Factor -POF	0	1	NA	NA	0,02	0	0,01
Usabilidad	Auto descripción	Medidas de boehm para auto descripción			0,3	0,2	0,5	0,625	0,41
		Quantity of comments (Por modulo)	0	1	NA	NA	NA	0,11	0,11
	Facilidad de aprendizaje	Effectiveness of the user documentation and/or help system	0	1	NA	NA	NA	0	0,00
		Effectiveness of the user documentation and/or help system in use	0	1	NA	NA	0,7	0,85	0,78

El nivel de aplicación fue medido por el número de métricas escogidas versus el número de métricas que pueden aplicarse por producto en cada empresa. El número total de métricas presentadas fue 15 y el número total de métricas depende del artefacto que la empresa suministra para realizar la evaluación.

- Para la empresa Latin Business Ltda., el número total de métricas aplicadas fue 13. Por tanto, el porcentaje de aplicación es de aproximadamente 87 por ciento, lo cual es considerable teniendo en cuenta la cercanía al máximo. Según los resultados obtenidos pueden tenerse a consideración los siguientes resultados:

Atributo	Concepto
Confiabilidad	Presenta un nivel de confianza bajo debido a que no hace claridad en relación al ocultamiento (privado y protegido) de métodos y atributos. Además presenta un acoplamiento (relación entre objetos: clases, atributos y métodos) muy alta, lo cual puede ser riesgoso en el momento de hacer cambios.
Funcionalidad	Algunas clases presentan un número moderadamente elevado de atributos, sin embargo esta dentro de lo normal. El número de clases para el modulo de controladores esta en un número adecuado. A pesar de poseer la habilidad de Reutilización en métodos hay que tener cuidado pues el acoplamiento es muy alto. En relación a control de acceso y encriptación de los datos tienen que estar en un buen nivel con posibilidad de mejora.
Mantenibilidad	En este momento el Mantenimiento del producto puede verse comprometido debido a que hay niveles más altos de acoplamiento y relaciones entre los objetos de diseño. Esto en

	parte porque la capacidad de Encapsulamiento es baja.
Usabilidad	En relación a la documentación de código esta que en la mayor parte de código no son encontrados comentarios, o son mínimos. Las Funcionalidades en relación a los requerimientos han sido cumplidas para uso del usuario pero aun faltan algunas.

- Para la empresa SITI Ltda., el número total de métricas aplicadas fue 15, esto quiere decir que en el producto suministrado fue posible identificar la totalidad de aspectos que proveen las métricas encontradas, y además es un posible indicio de buenas prácticas encontradas en dicho producto.

Atributo	Concepto
Confiabilidad	El modulo presenta una protección adecuada de sus atributos en las clases, sin embargo el ocultamiento de los métodos no es considerablemente bueno; a pesar de esto, muestra un acoplamiento entre clases relativamente bajo, lo cual favorece la Reutilización de código.
Funcionalidad	Algunas de las clases tienen un número elevado de atributos y métodos.
Mantenibilidad	Para realizar Mantenimiento del software, puede presentarse problemas debido al bajo ocultamiento de los métodos y el uso de atributos compartidos entre clases, sin embargo existe la posibilidad de mejorar esto debido a que el acoplamiento no es muy alto.
Usabilidad	En relación a la documentación de código, se encuentra que la empresa posee una herramienta de auto documentación lo cual hace favorable la auto-descripción; sin embargo el porcentaje de comentarios es muy bajo. Por otro lado en referencia a la documentación para usuario se tiene que cuando ésta existe es medianamente efectiva.

- Para la empresa Input Technologies Ltda., el número total de métricas aplicadas fue 9. Por tanto, el porcentaje de aplicación es de aproximadamente 60 por ciento, indicando que algunas de las prácticas que sugieren las métricas no fueron realizadas en el producto suministrado.

Atributo	Concepto
Confiabilidad	Presenta un nivel de confianza limitado debido a que el ocultamiento (privado y protegido) de métodos y atributos es moderadamente bajo, esto se puede ver reflejado en riesgos de violación de integridad. En relación al acoplamiento (relación entre objetos: clases, atributos y métodos) se encuentra en un buen nivel, lo cual implicara mayor Facilidad al momento de realizar cambios y reutilizar código

Funcionalidad	En relación al tamaño (tomado desde atributos y metodos) se encuentra en un buen nivel lo cual implica una menor Complejidad a la hora de rehusar y hacer cambios, el valor esta dentro de lo normal. El numero de clases para el modulo de Actas esta en un nivel bueno. En relación a control de acceso se ha encontrado un nivel sobresaliente, lo cual da un buen indicio de seguridad de los datos de usuarios; y para encriptación de los datos se tiene que debido a que no Aplica es riesgoso para la herramienta en caso de un ataque mal intencionado.
Mantenibilidad	El modulo de actas puede ser fácil de mantener y reutilizar debido a que no existe un acoplamiento elevado. Tener e cuenta el ocultamiento de los datos pues al ser bajo, se puede presentar problemas.
Usabilidad	La Usabilidad se midió en términos de auto descripción de código y no de funcionamiento de interfaz. En relación al los comentarios usados en el código se encontró niveles muy bajos, lo cual es poco deseable pues no favorece el Mantenimiento o los cambios al código fuente.

- Para la empresa Seratic Ltda., el número total de métricas aplicadas fue 11. Por tanto, el porcentaje de aplicación es de aproximadamente 73 por ciento, lo cual es considerable teniendo en cuenta la cercanía al máximo.

Atributo	Concepto
Confiabilidad	La Confiabilidad se ve dividida entre el ocultamiento (privado y protegido) de los atributos y los métodos, en relación a atributos posee un buen nivel de ocultamiento; sin embargo los métodos no, esto puede ser riesgoso, pues se ven implicados en problemas de seguridad. Por otro lado el acoplamiento (relación entre objetos: clases, atributos y métodos) se encuentra en un buen nivel, lo cual implicara mayor Facilidad al momento de realizar cambios a nivel de clases y reutilizar código donde ellas se comunican, sin embargo debido a que algunas clases tienen demasiados atributos y métodos puede tornarse algo tedioso encontrar errores y cambios.
Funcionalidad	La Funcionalidad en relación a la seguridad manejada para el acceso está en un muy buen nivel, lo que implica que los usuarios no podrán acceder a lugares no autorizados de la aplicación. Un aspecto que se encontró también relacionado a Funcionalidad fue el tamaño de las clases, que para el módulo fue demasiado grande, esto puede causar problemas de Mantenimiento.
Mantenibilidad	Se encuentra en un estado bueno pero con posibilidad de mejora para la ocultación de los datos. Un problema encontrado para la Facilidad de Mantenimiento fue el tamaño del diagrama, en relación con atributos y métodos, pues el número de ellos fue demasiado elevado, lo cual puede ser problemático en caso de querer reutilizar el módulo.
Usabilidad	La Usabilidad se midió en términos de auto descripción de código y no de funcionamiento de interfaz. En relación a los comentarios usados en el código y las prácticas que se deben usar al codificar

	o comentar código, se encontró niveles muy bajos, lo cual es poco deseable pues no favorece el Mantenimiento o los cambios a los que pueda haber lugar.
--	---

Puede observarse que si fue posible aplicar el compendio escogido a partir del marco conceptual, pero requiere que la persona que haga uso de los elementos del marco, tenga una instrucción previa y disposición temporal suficiente.

Después de aplicadas las métricas puede verse que para cada artefacto en las empresas existió un porcentaje en el cual el compendio pudo ser aplicado, logrando un promedio de aplicación de 80%, mostrando que en su mayoría los elementos pueden ser aplicados.

ANEXO R: ENCUESTA DE SATISFACCIÓN

El presente anexo describe la encuesta realizada a un conjunto de MiPyMEs Desarrolladoras de Software de Popayán y Pasto. La razón de la creación de la encuesta obedece a la necesidad de conocer las opiniones de las MiPyMEs Desarrolladoras de Software en relación al trabajo realizado con ellos a lo largo de este proyecto. Primero presenta el formato de la encuesta realizada y posteriormente los resultados obtenidos en cada una de las empresas. Fue realizada durante el mes de noviembre de 2009

R1. Diseño de la Encuesta

ENCUESTA DE SATISFACCIÓN

Responsables:

Mag. Jorge Jair Moreno Chaustre

Estudiantes:

Liliam Paola Bolaños Rengifo

Manuel Alejandro Navia Porras

Objetivo: Conocer las opinión de las empresas frente a la información presentada de parte de los responsables acerca de los documentos antes presentados

DATOS DEL ENTREVISTADO	
Nombres Completos:	
Apellidos Completos:	
Correos Electrónicos:	
Organización:	
País:	
Ciudad:	
Fecha:	

POR FAVOR CONTESTE LAS SIGUIENTES PREGUNTAS MARCANDO EN LOS CASILLEROS DE ACUERDO CON LOS CRITERIOS INDICADOS

MUY INSATISFECHO 1 2 3 4 5 MUY SATISFECHO
 INSATISFECHO INSATISFECHO REGULAR SATISFECHO SATISFECHO

1 ¿Cuál es su nivel de satisfacción con la información presentada en cuanto a:

	1	2	3	4	5
: Comprensibilidad?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
: Sencillez?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
: Usabilidad?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2 ¿Cuál es su nivel de satisfacción en cuanto a la calidad de la información presentada?



	1	2	3	4	5
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Gracias por su atención y colaboración.

Firma del entrevistado

Firma del entrevistador

R2. Resultados de la Empresa SERATIC

ENCUESTA DE SATISFACCIÓN

Responsables:
MAG. Jorge Jair Moreno Chaustre

Estudiantes:
Liliam Paola Bolaños Rengifo
Manuel Alejandro Navia Porras

Objetivo: Conocer las opinión de las empresas frente a la información presentada de parte de los responsables acerca de los documentos antes presentados

DATOS DEL ENTREVISTADO	
Nombres Completos:	Diego Ivan
Apellidos Completos:	Chamorro Salas
Correos Electrónicos:	Diego_chamorro@seratic.com
Organización:	Seratic Ltda.
País:	Colombia
Ciudad:	Popayan
Fecha:	9 diciembre de 2009]

POR FAVOR CONTESTE LAS SIGUIENTES PREGUNTAS MARCANDO EN LOS CASILLEROS DE ACUERDO CON LOS CRITERIOS INDICADOS

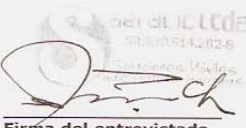
MUY INSATISFECHO	INSATISFECHO	REGULAR	SATISFECHO	MUY SATISFECHO
1	2	3	4	5

1 ¿Cuál es su nivel de satisfacción con la información presentada en cuanto a:

	1	2	3	4	5
: Comprensibilidad?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
: Sencillez?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
: Usabilidad?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

2 ¿Cuál es su nivel de satisfacción en cuanto a la calidad de la información presentada?

	1	2	3	4	5
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>



Firma del entrevistado

Gracias por su atención y colaboración.

Firma del entrevistador

R3. Resultados de la Empresa INPUT TECHNOLOGIES



ENCUESTA DE SATISFACCIÓN

Responsables:

MAG. Jorge Jair Moreno Chaustre

Estudiantes:

Liliam Paola Bolaños Rengifo

Manuel Alejandro Navia Porras

Objetivo: Conocer las opinión de las empresas frente a la información presentada de parte de los responsables acerca de los documentos antes presentados

DATOS DEL ENTREVISTADO	
Nombres Completos:	RICARDO ANDRÉS
Apellidos Completos:	LEDEZMA CHAVEZ
Correos Electrónicos:	ricardo.ledezma@input.com.co; rledezma@gmail.com
Organización:	INPUT TECHNOLOGIES LTDA.
País:	COLOMBIA
Ciudad:	POPAYÁN
Fecha:	2009-11-13

POR FAVOR CONTESTE LAS SIGUIENTES PREGUNTAS MARCANDO EN LOS CASILLEROS DE ACUERDO CON LOS CRITERIOS INDICADOS

MUY INSATISFECHO INSATISFECHO REGULAR SATISFECHO MUY SATISFECHO

1 2 3 4 5

1 ¿Cuál es su nivel de satisfacción con la información presentada en cuanto a:

	1	2	3	4	5
: Comprensibilidad?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
: Sencillez?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
: Usabilidad?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

2 ¿Cuál es su nivel de satisfacción en cuanto a la calidad de la información presentada?

1	2	3	4	5
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Felicitaciones, muy buen trabajo.

Gracias por su atención y colaboración.

Ricardo Andrés Ledezma
Firma del entrevistado

Firma del entrevistador

R4. Resultados de la Empresa LATIN BUSINESS

ENCUESTA DE SATISFACCIÓN

Responsables:

MAG. Jorge Jair Moreno Chaustre

Estudiantes:

Liliam Paola Bolaños Rengifo

Manuel Alejandro Navia Porras

Objetivo: Conocer las opinión de las empresas frente a la información presentada de parte de los responsables acerca de los documentos antes presentados

DATOS DEL ENTREVISTADO	
Nombres Completos:	John Jairo
Apellidos Completos:	Ortiz Canchala
Correos Electrónicos:	jjotiz1@parquesoftpasto.com , jjortiz1@gmail.com
Organización:	Latin Business Ltda.
País:	Colombia
Ciudad:	Pasto
Fecha:	09-11-09

POR FAVOR CONTESTE LAS SIGUIENTES PREGUNTAS MARCANDO EN LOS CASILLEROS DE ACUERDO CON LOS CRITERIOS INDICADOS

MUY INSATISFECHO INSATISFECHO REGULAR SATISFECHO MUY SATISFECHO
 1 2 3 4 5

1 ¿Cuáles su nivel de satisfacción con la información presentada en cuanto a:

	1	2	3	4	5
: Comprensibilidad?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
: Sencillez?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
: Usabilidad?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

2 ¿Cuáles su nivel de satisfacción en cuanto a la calidad de la información presentada?

1	2	3	4	5
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Agradecemos el trabajo realizado y quedamos atentos a la continuación de este proyecto y ojala podamos recibir En un futuro el fruto final de las investigaciones que están llevando a cabo, con el propósito de aplicar sus Sugerencias a nuestro desarrollo de software.



Firma del entrevistado

Gracias por su atención y colaboración.

Firma del entrevistador

R5. Resultados de la Empresa SITI



ENCUESTA DE SATISFACCIÓN

Responsables:

MAG. Jorge Jair Moreno Chaustre

Estudiantes:

Liliam Paola Bolaños Rengifo

Manuel Alejandro Navia Porras

Objetivo: Conocer las opinión de las empresas frente a la información presentada de parte de los responsables acerca de los documentos antes presentados

DATOS DEL ENTREVISTADO	
Nombres Completos:	GELBER ORLANDO
Apellidos Completos:	MORAN SILVA
Correos Electrónicos:	gelbermoran@siticol.com
Organización:	Soluciones Integrales en Tecnologías de la Información S.I.T.I. Ltda.
País:	Colombia
Ciudad:	Pasto
Fecha:	10 de Noviembre de 2009

POR FAVOR CONTESTE LAS SIGUIENTES PREGUNTAS MARCANDO EN LOS CASILLEROS DE ACUERDO CON LOS CRITERIOS INDICADOS

MUY INSATISFECHO	INSATISFECHO	REGULAR	SATISFECHO	MUY SATISFECHO
1	2	3	4	5

1 ¿Cuál es su nivel de satisfacción con la información presentada en cuanto a:

	1	2	3	4	5
: Comprensibilidad?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
: Sencillez?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
: Usabilidad?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

2 ¿Cuál es su nivel de satisfacción en cuanto a la calidad de la información presentada?

1	2	3	4	5
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Gracias por su atención y colaboración.


Firma del entrevistado

Firma del entrevistador

ANEXO S: INFORME

Informe: Encuesta para el diagnóstico respecto a prácticas del producto software

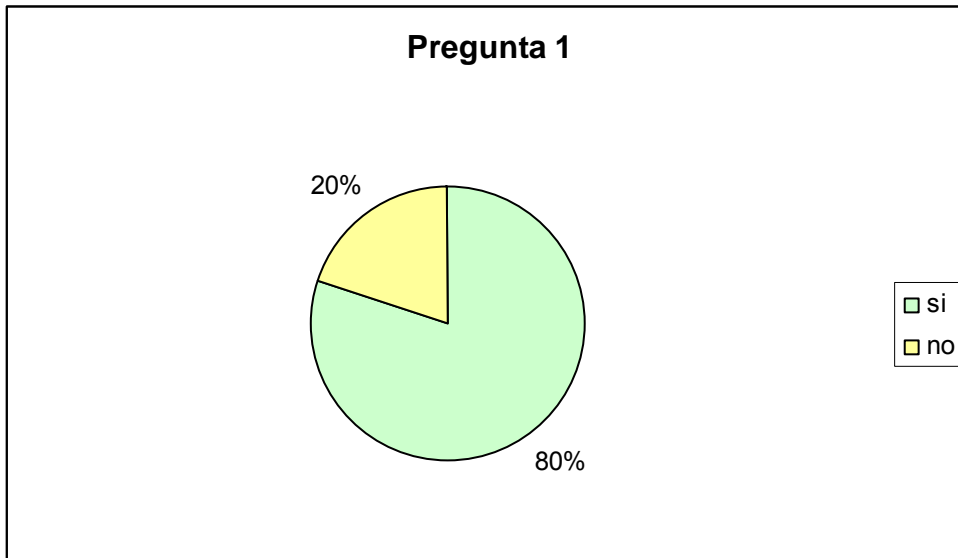
Objetivo: Conocer las prácticas relacionadas con el uso de métricas orientadas hacia la valoración del producto software con la encuesta realizada en la primera semana del mes de septiembre de 2008 (ver anexo) a las empresas de desarrollo de software de PARQUESOFT y estudiantes de últimos semestres de Ingeniería de Sistemas de la Universidad del Cauca.

Población: estudiantes de últimos semestres de Ingeniería de Sistemas de la Universidad del Cauca, Funcionarios de empresas de desarrollo de software en Popayán Cauca

Muestra: 15 personas

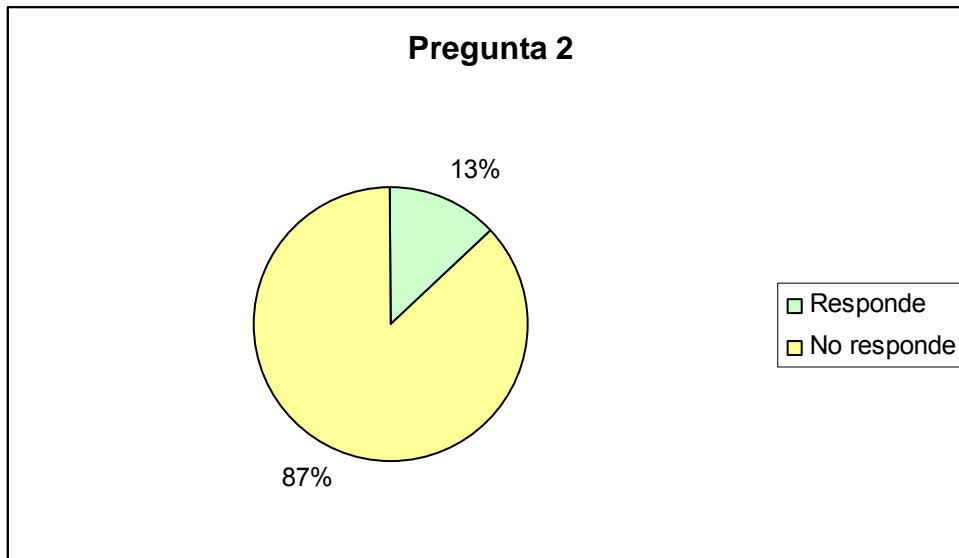
Resultado por pregunta

1. ¿Sabe usted qué es una métrica de software?

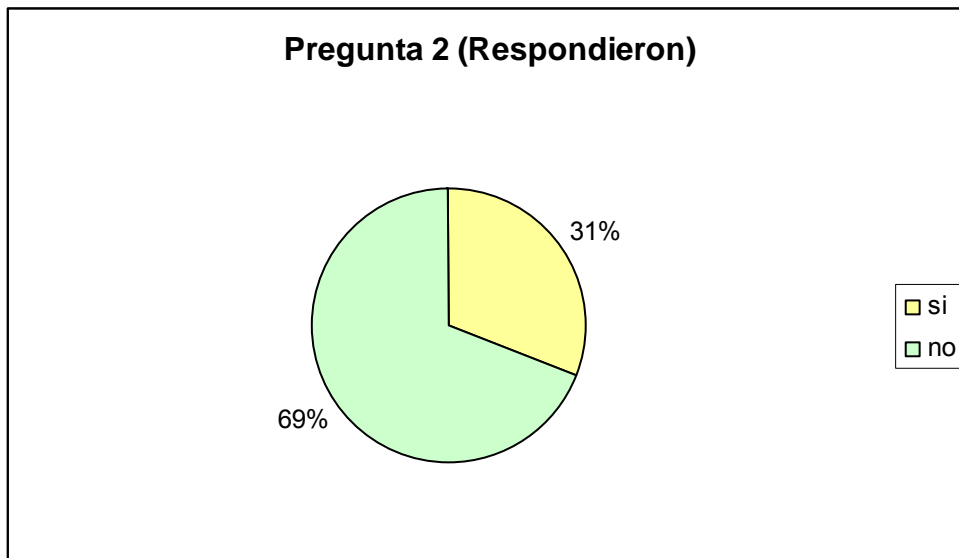


Del total de personas encuestadas, se obtuvo que el 80% conoce o tiene idea de que es una métrica de software; mientras que el 20% no está familiarizado con el término.

2. ¿Utiliza métricas para valorar la calidad de su producto software?

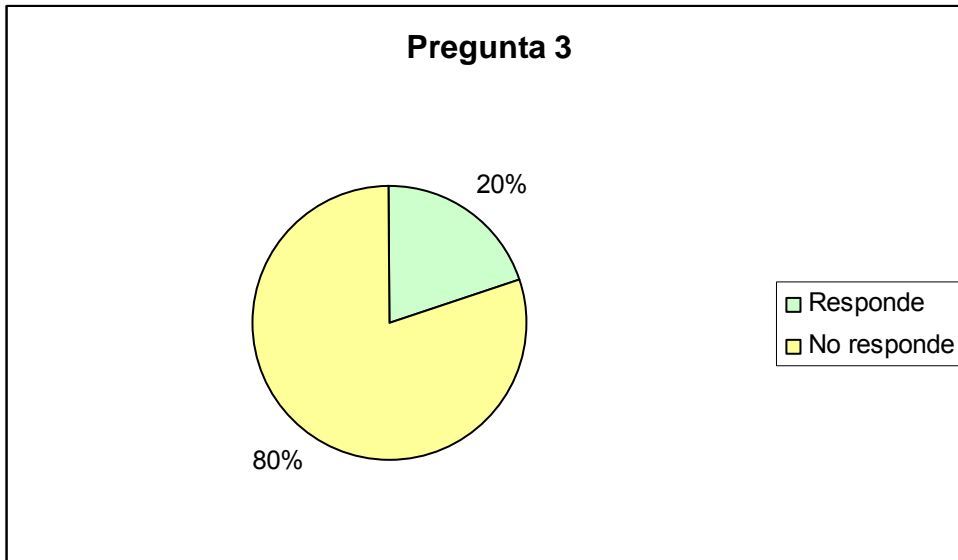


Del total de personas encuestadas, el 87% no respondió a esta pregunta; mientras que el 13% respondió de la siguiente manera:

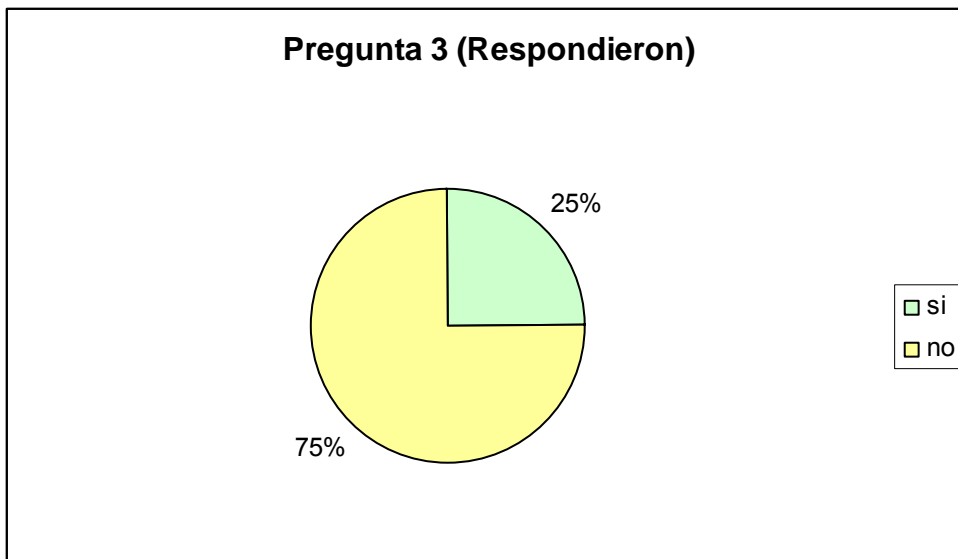


Del total de personas encuestadas que respondieron a esta pregunta, se obtuvo que el 31% ha utilizado métricas para valorar la calidad de su producto software; mientras que el 69% no las ha utilizado.

3. ¿Conoce de la existencia de herramientas que calculan o extraen métricas de software a través del código fuente?

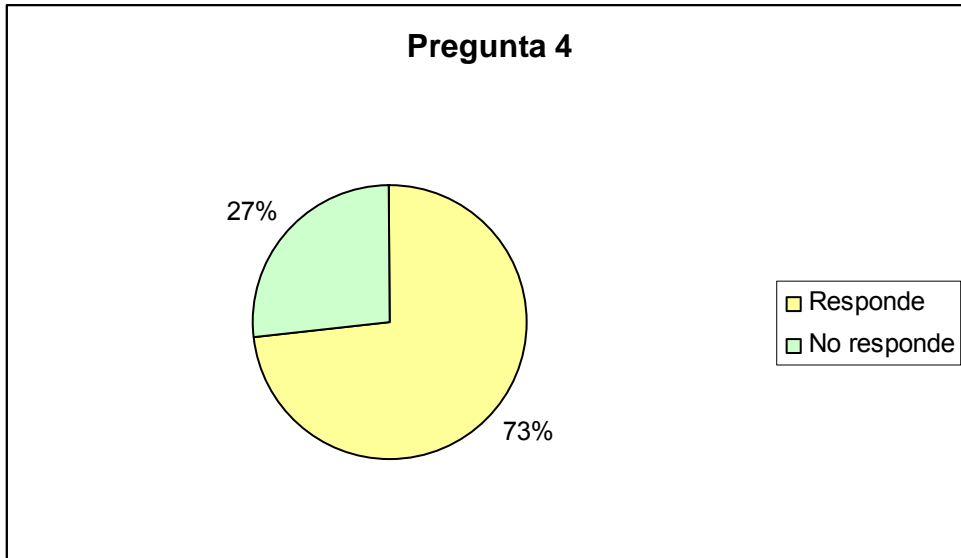


Del total de personas encuestadas, el 80% no respondió a esta pregunta; mientras que el 20% respondió de la siguiente manera:

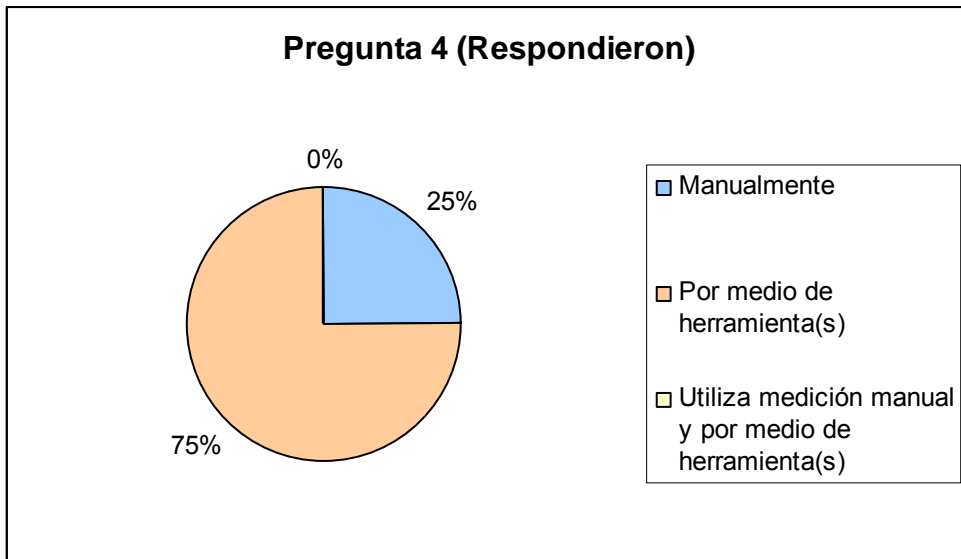


Del total de personas encuestadas que respondieron a esta pregunta, se obtuvo que el 25% conoce y utiliza herramientas que extraen métricas de software a través del código fuente; mientras que el 75% no las conoce y/o no las ha utilizado.

4. Si utiliza las métricas de software, ¿cómo las calcula?

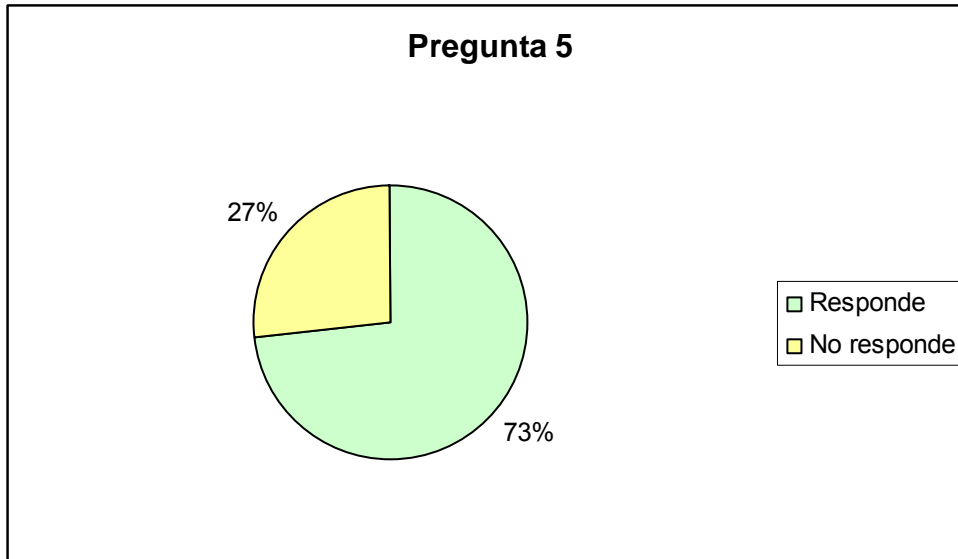


Del total de personas encuestadas, el 27% no respondió a esta pregunta; mientras que el 73% respondió de la siguiente manera:



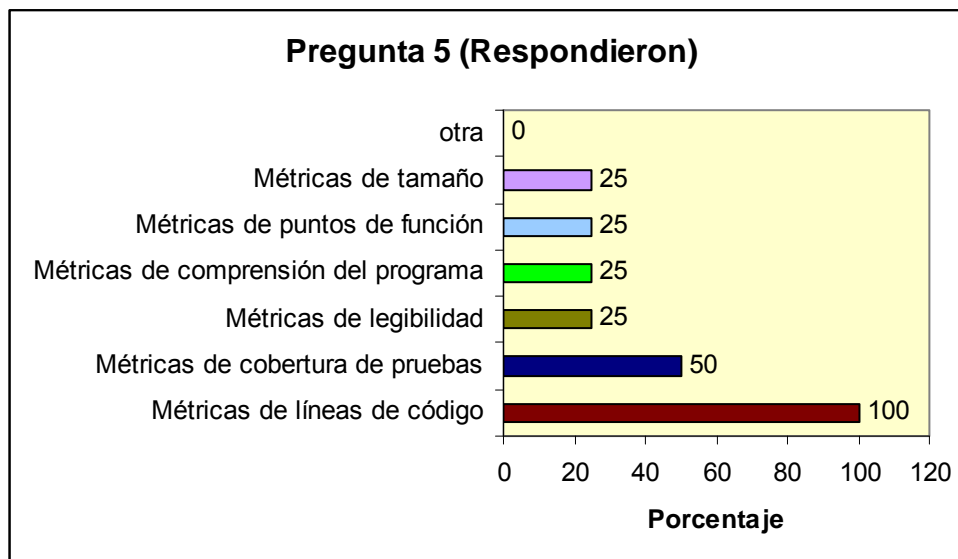
Del total de personas encuestadas que respondieron a esta pregunta, se obtuvo que el 25% al usar métricas de software las calcula de manera manual, el 75% hace uso de herramientas, mientras que el 0% hace uso de los dos métodos para hacer el cálculo.

5. Si utiliza la medición manual para lo anterior, entonces:



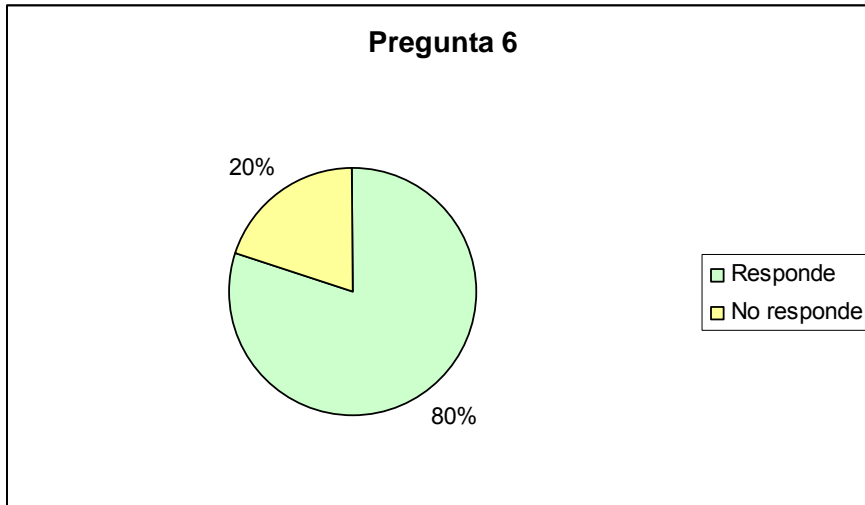
Del total de personas encuestadas, el 27% no respondió a esta pregunta; mientras que el 73% respondió de la siguiente manera por Descripción ó selección:

- **Describe:**
Algunos de los encuestados hacen uso de herramientas tales como visual studio o netbeans entre otras, sin embargo manifiestan que hay mayor beneficio en evaluar secciones de código que presenten prioridad en sus aplicaciones.
- **Selección:**

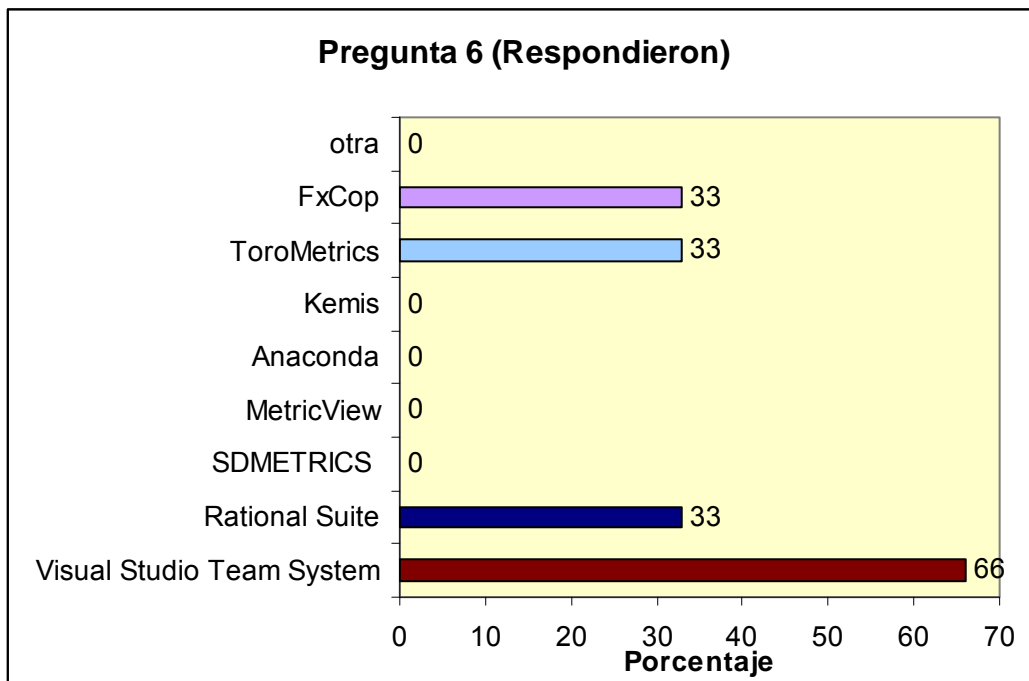


Del total de personas encuestadas que respondieron a esta pregunta, se obtuvo que el 25% ha usado métricas de tamaño, el 25% métricas de puntos de función, el 25% métricas de comprensión del programa, el 25% métricas de legibilidad, el 50% métricas de cobertura de pruebas y la totalidad de encuestados utiliza métricas de línea de código.

6. Si utiliza herramienta(s) para la medición de calidad del producto software, ¿Cuál(es) usa?



Del total de personas encuestadas, el 80% no respondió a esta pregunta; mientras que el 20% respondió de la siguiente manera:

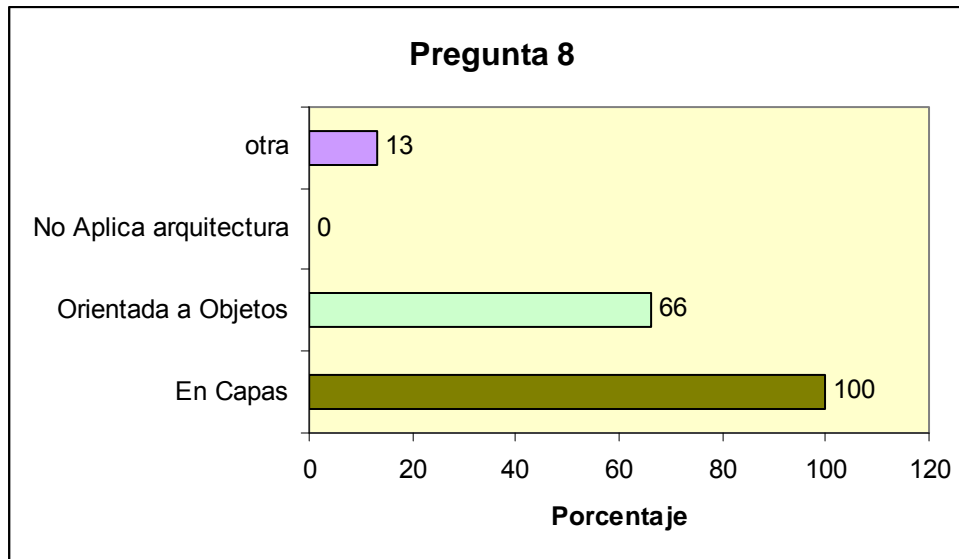


Del total de personas encuestadas que respondieron a esta pregunta, se obtuvo que el 33% ha usado ToroMetrics, el 33% FxCop, el 33% Rational Suite, el 66% Visual Studio Team System; las restantes no han sido utilizadas o no son conocidas.

7. En sus desarrollos, ¿conoce y Aplica el paradigma orientado a objetos?

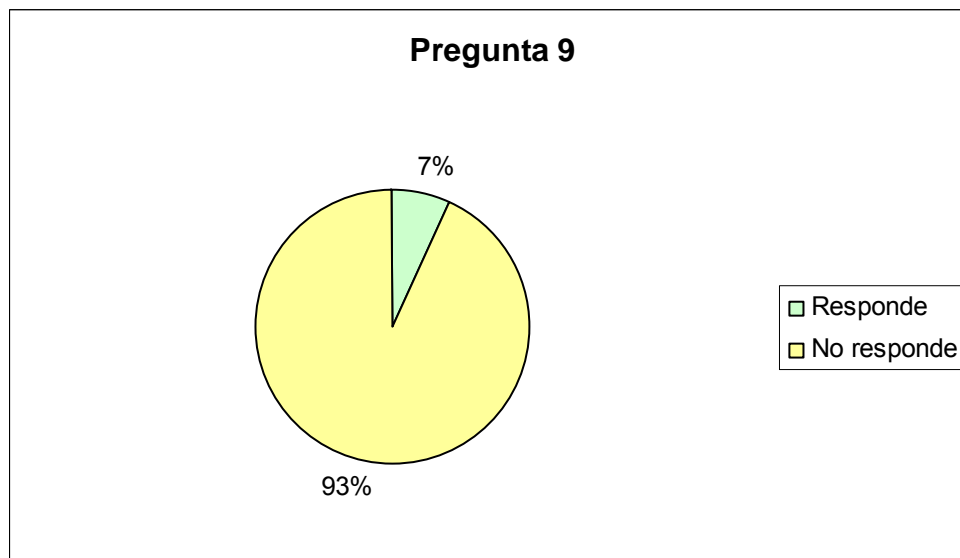
La totalidad de los encuestados manifestaron que conocen y aplican el paradigma Orientado a Objetos.

8. ¿Aplica alguna arquitectura software en sus proyectos?

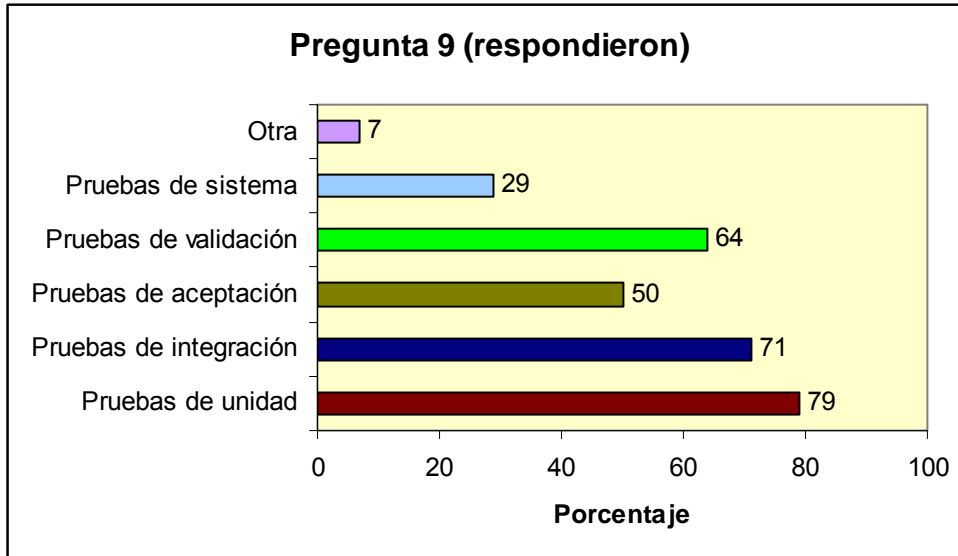


Del total de personas encuestadas, se obtuvo que el 66% utiliza la arquitectura Orientada a Objetos, el 100% arquitectura en Capas y el 13% utiliza otra arquitectura.

9. ¿Qué técnicas o estrategias utiliza para detectar los errores en su producto?



Del total de personas encuestadas, el 93% no respondió a esta pregunta; mientras que el 7% respondió de la siguiente manera:



Del total de personas encuestadas que respondieron a esta pregunta, se obtuvo que el 29% utiliza pruebas de sistema, el 64% pruebas de validación, el 50% pruebas de aceptación, el 71% pruebas de integración, el 79% pruebas de unidad y un 7% utiliza otro tipo de pruebas.

10. Comentarios y sugerencias.

Para los encuestados que dieron respuesta a este punto es importante lo siguiente:

- La calidad del producto software.
- Como valorar dicha Calidad.
- Conocer, entender y aplicar los procesos de valoración de calidad del producto software.
- La competitividad de sus productos en el ámbito nacional y global.

CONCLUSIONES

Entre las conclusiones que podemos mencionar se encuentran las siguientes:

- Falta completitud en la información que los encuestados tienen acerca de la valoración de calidad para el producto software. En algunas ocasiones se conoce la terminología pero no el significado.
- Existe en conocimiento mínimo acerca de las técnicas y herramientas que se pueden utilizar en la valoración de la calidad de un producto software o no se tiene acceso a ellas. Sin embargo aquellos que si hacen uso de estos recursos prefieren las herramientas para la valoración.

- En la encuesta se encontró que varios de los encuestados a pesar de tener conocimiento de calidad de software, no lo aplican de manera adecuada o no lo utilizan.
- La totalidad de los encuestados manifiestan conocer y aplicar el paradigma orientado a objetos
- Se encontró que la totalidad de los encuestados utiliza algún tipo de arquitectura de software en sus proyectos.
- Existe un número mínimo de encuestados que realizan pruebas al producto software que desarrollan.

ANEXO T: PLANTILLAS PARA LA EVALUACIÓN DEL PRODUCTO SOFTWARE

En el momento que desea aplicarse el marco conceptual, es propuesto el uso de dos plantillas, con el fin de llevar de una forma ordenada el manejo de los atributos, métricas y heurísticas presentados en el marco. Para este punto son propuestas dos plantillas una para la evaluación y otra para las observaciones a seguir. En cada uno de los casos es recomendable el manejo de versiones, y cada plantilla debe ser usada por cada artefacto, iteración o incremento del producto software que desee evaluarse.

T1. *Plantilla 1: Selección de atributos, sub – atributos y métricas.*

La plantilla de selección de atributos, sub – atributos y métricas, fue pensada como una herramienta para registrar los valores a medir en un artefacto del producto software, y tener sus resultados, en caso de que los últimos no sean favorables presentar recomendaciones.

Plantilla: Atributo, sub atributos y métricas						
Atributo	Sub-atributo	Métrica	V. Mínimo	V. Máximo	V. Evaluación	Recomendación
A1	SA1	M1	--	--	--	--
		M2	--	--	--	--
A2	SA2	M3	--	--	--	--
		M4	--	--	--	--
...	--	--	--	--
		...	--	--	--	--
An	SAn	Mn-1	--	--	--	--
		Mn	--	--	--	--

A: Atributo de calidad que desea valorarse. Perteneciente a las capas 1 o 2 del marco.

SA: sub atributo relacionado al atributo de capa 1 o 2 escogido. Perteneciente a capa 3 ya sea compartido o exclusivo.

M: métrica seleccionada para medir el nivel de un sub-atributo(s) previamente escogido(s). Pertenecientes a la capa 4 del marco.

- V. Mínimo: Valor mínimo del resultado de la métrica
- V. Máximo: Valor máximo del resultado de la métrica
- V. Evaluación: valor arrojado después de aplicar la métrica.
- Recomendación: en caso de ser necesaria, escribir la recomendación para lograr que la métrica adquiera un valor deseable en la evaluación.

T2. Plantilla 2: Selección de heurísticas y resultados de aplicación.

La plantilla de selección de heurísticas y resultados de aplicación, fue pensada como una herramienta para llevar un registro de cuáles son las heurísticas seleccionadas y que resultados son obtenidos después de su aplicación. Sirve esencialmente para saber cuándo aplicar las heurísticas y en qué casos dan mejores resultados.

Plantilla: Atributo y heurísticas				
Atributo	Heurística	Aplicada (si/no)	Fecha de aplicación	Resultados
A1	H1	--	--	--
	H2	--	--	--
...	...	--	--	--
	...	--	--	--
An	Hn-1	--	--	--
	Hn	--	--	--

A: Atributo de calidad que desea valorarse. Perteneciente a las capas 1 o 2 del marco.

H: Heurística relacionada al atributo escogido, normalmente plantea una observación que desea tenerse en cuenta para lograr un mejor Desempeño del atributo.

- Aplicada (si/no): responde si realmente la heurística fue aplicada a determinado artefacto
- Fecha de aplicación: registra la fecha en la cual empezó a aplicarse la heurística
- Resultados: registra resultados ya sean positivos o negativos, además de puntos que considera deberían mejorarse.

ANEXO U: REFERENCIAS

Este anexo muestra las referencias bibliografías usadas a lo largo del proyecto, además de mostrar las fuentes donde fueron obtenidas

No.	TITULO	AUTOR (ES)	DISPONIBLE EN	OTROS
1	Ingeniería Del Software, Un Enfoque Práctico	PRESSMAN, Roger	http://www.mcgraw-hill.es	McGraw - Hill. 5ª Edición, 2002
2	Sistemas de Gestión de calidad. Fundamentos y Vocabulario NTC – ISO 9000	ICONTEC Instituto Colombiano de Normas Técnicas y Certificación	http://www.iso.org/iso	División de publicaciones. 2000. 37 p. Bogotá
3	Software Quality Attributes, with source code metrics	JETTER, Andreas	http://www.inf.usi.ch/	Assessing. University of Zurich. 2006.
4	Estudio Comparativo De Los Modelos Y Estándares De Calidad Del Software.	SCALONE, Fernanda	www.fi.uba.ar/laboratorios/lsi/scalone-tesis-maestria-ingenieria-en-calidad.pdf	Universidad Tecnológica Nacional, Facultad Regional Buenos Aires. 2006
5	Modelos de Calidad Web. Clasificación de Métricas	SALANOVA, Pilar	http://www.issi.uned.es/CalidadSoftware/Noticias/PFC_1.doc	Escuela Técnica Superior De Ingeniería Informática. 2006
6	Compilación de un Modelo para Evaluar Atributos de Calidad en Productos Software”	MORENO, Jair. ANDRADE, Hugo. BOLAÑOS, Liliam	http://enlaceinformatico.unicauca.edu.co/docs/enlaceinformatico@unicauca.edu.co_Elv06e01a09.pdf	Revista Enlace Informático, Universidad Del Cauca. 2007
7	Marco De Referencia	FERNÁNDEZ,	Biblioteca Serrano, Universidad del Cauca	Universidad Del

No.	TITULO	AUTOR (ES)	DISPONIBLE EN	OTROS
	Centrado En La Arquitectura Para La Mejora De Características De Usabilidad En El Desarrollo De Aplicaciones Web Construidas Por MIPYMES	Verónica, HORMIGA, María, TULANDE, Aleyda		Cauca. 2008
8	Integración De Calidad Y Experiencia En El Desarrollo De Interfaces De Usuario Dirigidos Por Modelos	MONTERO, Francisco	http://www.isys.ucl.ac.be/bchi/publications/Ph.D.The ses/Montero-PhD2005.pdf	Universidad De Castilla La Mancha. 2005.
9	Informe: Encuesta para el diagnóstico respecto a prácticas del producto software.	MORENO, Jair. BOLAÑOS, Liliam. NAVIA, Manuel		Septiembre, 2008
10	Handbook of Statistics 7: Quality Control and Reliability.	KRISHNAIAH, Paruchuri. VAN DIJK, Nicolette	www.amazon.com	Elsevier Science Publishers. North Holland, Ámsterdam. 1988.
11	Ingeniería Del Software	SOMMERVILLE, Ian	http://www.institutopascal.edu.ar/Alumnos/AsignaturasAnalistasDeSistemas/1ano/ArquitecturaDelComputador/IngenieriaSoft.pdf	Addison Wesley. 7ª Edición. 2005
12	Método Del Desarrollo Arquitectónico en Grupo	GONZALES, Moisés	http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/gonzalez_d_h/	Instituto Politécnico Internacional, México. 2006
13	Un Marco Conceptual para -la	OLSINA, Luis. BERTOA, Manuel,	http://www.lcc.uma.es/~av/Publicaciones/02/MarcoConceptualJISBD02.pdf	2002

No.	TITULO	AUTOR (ES)	DISPONIBLE EN	OTROS
	Definición y Explotación de Métricas de Calidad.	LAFUENTE, Guillermo		
14	Software Architecture in practice	BASS, Len, et. al	www.amazon.com	Addison Wesley. 2ª Edición. 2003
15	Information Technology. Part 1, Part 2, Part 3 and Part 4.	ISO/IEC 9126-2001	http://www.iso.org/iso	The International Standard Organization. 2002
16	IEEE 1061-1998, IEEE Standard for a Software Quality, Metrics Methodology	Software Engineering Standards Committee of the IEEE Computer Society	http://standards.ieee.org/	Software Engineering Standards Committee of the IEEE Computer Society.
17	Characteristics of Software Quality	BOEHM, Barry, BROWN, John. LIPOW, Myron. MACLEOD, Gordon. MERRIT, Michael	http://www.amazon.com	North-Holland, N.Y., 1978
18	Factors in software quality	McCALL, Jim A. RICHARDS, Paul K. WALTERS, Gene F	http://www.ntis.gov/	Nat'l Tech. Information Service, Vol 1, Vol 2 y Vol 3 , 1977
19	De Calidad Enfocados A La Usabilidad, Aplicados En Los Procesos De Desarrollo De Sistemas De Información	CALDERÓN, Shlomi	http://alarcos.inf-cr.uclm.es/doc/cmsi/trabajos/Shlomi%20Calderon%20-%20Modelos%20de%20Calidad%20Enfocados%20a%20la%20Usabilidad%20-%20Doc.pdf	Modelos. Universidad De Castilla-La Mancha, Escuela Politécnica Superior
20	Software Metrics: Establishing a	GRADY, Robert. CASWELL,	http://www.amazon.com	Prentice-Hall, 1987.

No.	TITULO	AUTOR (ES)	DISPONIBLE EN	OTROS
	Company-Wide Program	Deborah		
21	Mejora de la calidad del software en el entorno de microempresas de TI	CABALLERO, Edgar	http://is.ls.fi.upm.es/doctorado/Trabajos20062007/Caballero.pdf	Universidad Politécnica de Madrid. 2007
22	Software Quality: Definitions and Strategic Issues. Advanced Research Module	FITZPATRICK, Ronan	http://www.comp.dit.ie/rfitzpatrick/papers/quality01.pdf	Staffordshire University, School of Computing Report. April 1996.
23	IEEE Standard Glossary of Software Engineering Terminology	IEEE 610.12 -1990	http://standards.ieee.org/	The Institute of Electrical and Electronics Engineers
24	Metrics and Models in Software Quality Engineering	KAN, Stephen	http://www.amazon.com	Addison Wesley. 2ª Edición. 2002.
25	Documentación sobre SATC	Portal de la NASA	http://satc.gsfc.nasa.gov	Abril de 2008
26	A Model For Software Product Quality	DROMEY, Geoff	http://www.computer.org/portal/web/csdl/doi/10.1109/32.345830	Software Quality Institute Griffith University. Australia. 1994
27	ISO/IEC 14598-1. Information Technology - Software Product Evaluation – Parte 1: General overview	The International Standard Organization	http://www.iso.org/iso	1995
28	Calidad en modelos conceptuales: Un análisis multidimensional de	MARÍN, Beatriz, CONDORI-FERNÁNDEZ, Nelly, PASTOR,	http://www.aemes.org/rpm/descargar.php?volumen=4&numero=4&articulo=8	Vol. 4. 2007

No.	TITULO	AUTOR (ES)	DISPONIBLE EN	OTROS
	modelos cualitativos basados en la ISO 9126. RPM-AEMES	Oscar		
29	Involving a corporate software quality assessment exercise: A migration path to ISO/IEC 9126.	CÔTÉ, Marc-Alexis, SURYN, Witold, MARTIN, Robert, LAPORTE, Claude	http://profs.logti.etsmtl.ca/claporte/English/Publications/index.html	Involving, Vol. 6, N° 3. 2004
30	Providing a framework for effective software quality assessment: A first step in automating assessments	MARTIN, Robert, SHAFER, Lawrence	http://www.mitre.org/work/tech_transfer/pdf/assessment_automation.pdf	The First Annual Software Engineering & Economics Conference. MITRE Hayes Auditorium, Virginia. 1996
31	A Hierarchical Model for Quality Assessment of Object-Oriented Designs, doctoral dissertation	BANSIYA, Jagdish, DAVIS, Carl	http://www.computer.org/portal/web/csdl/abs/trans/ts/2002/01/e0004abs.htm	The University of Alabama in Huntsville, 1997.
32	The goal question metric approach	BASILI, Victor, CALDIERA, Gianluigi, ROMBACH, Dieter	http://www.cs.umd.edu/~basili/publications/technical/T100.pdf	Encyclopedia of Software Engineering, pp. 528-532, John Wiley & Sons, Inc. 1994.
33	Construcción de una herramienta para evaluar la calidad de un producto software	PIEDRAHITA, Sebastián	http://bdigital.eafit.edu.co/	Universidad EAFIT. Departamento de ingeniería de

No.	TITULO	AUTOR (ES)	DISPONIBLE EN	OTROS
				sistemas. 2007
34	Procesos de las aplicaciones web: Informe sobre la "calidad de las aplicaciones web"	ABRIL, Enrique	http://zarza.usal.es/~fgarcia/doctorado/iweb/05-07/Trabajos/CalidadAplicsWeb.pdf	
35	Estudio del portal de Castilla-lamanca.es	MORAGA, M ^a Ángeles, CALERO, Coral, MARTÍNEZ, Julián, POZUELO, Antonio, ESPADAS, María	http://www.iadis.net/dl/final_uploads/200508C015.pdf	Conferencia IADIS Ibero-Americana. 2005
36	Software quality attributes and trade-offs	BERANDER, Patrik, et. al.	http://www.bth.se/besq	Blekinge Institute of Technology. 2005
37	The SMPI model: A stepwise process model to facilitate software measurement process improvement along the measurement paradigms	DUMKE, Reiner, CUADRADO-GALLEGO, Juan, ABRAN, Alain	http://diglib.uni-magdeburg.de/Dissertationen/2007/renbraungarten.pdf	Otto-von-Guericke-Universität Magdeburg. 2007
38	Guía de mejores prácticas de calidad de producto	INTECO	http://www.inteco.es/Calidad_del_Software/descargas/guias/guia_de_mejores_practicas_de_calidad_de_producto	Instituto Nacional de Tecnologías de la Información. 2008
39	ISO/IEC 12207(E) / IEEE std 12207. International Standard – Systems and software engineering –	The International Standard Organization	http://ieeexplore.ieee.org , http://www.iso.org/iso	2008

No.	TITULO	AUTOR (ES)	DISPONIBLE EN	OTROS
	Software life cycle processes			
40	Classifying web metrics using the web quality model	CALERO, Coral, RUIZ, Julián, PIATTINI, Mario	http://www.emeraldinsight.com/Insight/viewContentItem.do?contentType=Article&contentId=1509084	Emerald, Online information review. Vol. 29, N° 3. Pg.: 227-248. 2005
41	Evaluación de la calidad de software en sistemas de información en internet	DAVILA, Leticia, MEJIA, Pedro	http://delta.cs.cinvestav.mx/~pmalvarez/davila-mejia.pdf	
42	ISO/IEC 25000 International Standard – Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE	The International Standard Organization.	http://www.iso.org/iso	2005
43	ISO/IEC 20000 International Standard – Information Technology – Service management- part 1: Specification	The International Standard Organization	http://www.iso.org/iso	2005
44	Relaciones humanas: Curso de comportamiento en la organización	SORIA, Víctor	http://www.monografias.com/trabajos5/relhuman/relhuman.shtml	Limisa. 2002
45	Reusable Component Engineering For Hard	CORNWELL, Peter	http://www.cs.york.ac.uk/ftpd/ftpdir/reports/98/YCST/04/YCST-98-04.pdf	University of York, UK. 1998.

No.	TITULO	AUTOR (ES)	DISPONIBLE EN	OTROS
	Real-Time Systems			
46	Modeling human mental processes	SIMON, Hebert.	http://portal.acm.org/citation.cfm?id=1460703	Proc. (AFIPS). Western Joint Computer Conference, Vol. 19, 1961
47	Herramientas para la gerencia del conocimiento	HENAO CALAD, Mónica	http://atlas.eafit.edu.co:8001/servlet/SBReadResourceServlet?rid=1175636219140_1024010182_2161	Universidad EAFIT. Medellín
48	An Overview of Object Oriented Design Heuristics	BASHAR, Muhammad	http://www.cs.umu.se/education/examina/Rapporter/BasharMolla.pdf	Department of Computer Science, Umeå University, Sweden. 2005
49	Definición, “Pyme, definición”	Portal de Micro, Pequeña y Mediana Empresa - MIPYMES - Mincomercio Colombia	http://www.mipymes.gov.co/pyme/home.asp	
50	Importancia de las MiPyMEs en las Aglomeraciones Empresariales. * Una estrategia para el desarrollo regional en Colombia	ALEMAN, Fernando	http://www.umng.edu.co/revcieco/2006/PDF%20de%20Corel/Importancia.pdf	
51	Marco Conceptual para la implantación de Gestión del Conocimiento en un Programa de Mejora de Procesos Software en MiPyMEs DS	CAPOTE, Johana, LLANTÉN, Julián	Biblioteca Serrano, Universidad del Cauca	Grupo de Investigación IDIS - Investigación y Desarrollo. 2008.

No.	TITULO	AUTOR (ES)	DISPONIBLE EN	OTROS
52	Principles of Software Engineering Management	GILB, Tom	http://www.amazon.com	Addison Wesley, 1987
53	Revisión Sistemática de Métricas de Diseño Orientado a Objetos	OLMEDILLA, Juan	http://is.ls.fi.upm.es/doctorado/Trabajos20042005/Olmedilla.pdf	Universidad Politécnica de Madrid, Facultad de Informática. 2005
54	Software product quality: theory, model, and practice	DROMEY, Geoff	http://www.sqi.gu.edu.au/docs/sqi/misc/SPQ-Theory.pdf	Griffin University. Australia, Brisbane, Technical Report. 1998
55	Product Software Quality	PANOVSKI, Gregor	http://www.cs.ru.nl/~marko/onderwijs/masterscripties/GregorPanovskiThesis.pdf	Techische Universiteit Eindhoven. 2008
56	Applying the ISO 9126 Quality Model to Test Specifications Exemplified for TTCN-3 Test Specifications	ZEISS, Benjamin, VEGA, Diana, SCHIEFERDECKE R, Ina, NEUKIRCHEN, Helmut, GRABOWSKI, Jens	http://www.swe.informatik.uni-goettingen.de/pubs/single_pub/index.php?lang=de&pub_nr=266	
57	Object-oriented designs knowledge: principles, heuristics and best practices	GARZAS, Javier, PIATTINI, Mario	http://www.amazon.com	IGI Global. Edición. 2006
58	Quint II, "Kwaliteit van software producten"	B. Van Zeist, P. Hendriks, R. Paulussen en J. Trienekens	http://www.serc.nl/quint-book	Kluwer Bedrijfswetenschappen, Deventer, ISBN 90-267-2430-6.
59	The Extended ISO Model of Software	Página de QUINT2	http://www.serc.nl/quint-book	

No.	TITULO	AUTOR (ES)	DISPONIBLE EN	OTROS
	Quality			
60	Using WQM for clasifying usability metrics	PENICHET, Victort, Calero, C, LOZANO, Maria, PIATINI, Mario	http://penichet.net/~penichet/images/stories/profesional/investigacion/docs/publicaciones/penichet_ICWI06.pdf	Proceedings of the IADIS International Conference. 2006; I.S.B.N.: 972-8924-19-4, Murcia, Spain; 05 Oct 2006.
61	Métricas Orientadas a Objetos. Technical Report, DPTOIA-IT-2001-002	VÁZQUEZ, Pedro, MORENO, María, GARCÍA, Francisco	http://tejo.usal.es/inftec/2001/DPTOIA-IT-2001-002.pdf	Departamento de Informática y Automática, Universidad de Salamanca. 2001
62	Object – Oriented Software Construction	MEYER, Bertrand	http://www.amazon.com	Prentice-Hall ISE Inc. 2ª Edición, 1991
63	Metrics in software quality assurance	GAFFNEY, Jhon Jr.	http://portal.acm.org/citation.cfm?id=809854	Portal ACM. 1991
64	Quality attributes for a component quality model	ÁLVARO, Alexandre, SANTANA DE ALMEIDA, Eduardo, ROMERO DE LEMOS MEIRA, Silvio	http://research.microsoft.com/~cszypers/events/WCOP2005/09%20-%20Alvaro.pdf	Technical communications, Kansas City Division. 1990
65	Software Quality Assurance Handbook. KCP-613-4118	Software Quality Assurance Group	http://www.ntis.gov/search/product.aspx?ABBR=DE91000060	
66	Code Complete: A Practical Handbook of Software Construction	McCONNELL, Steven	http://cc2e.com/	Microsoft Press. 2ª Edición, 2004

No.	TITULO	AUTOR (ES)	DISPONIBLE EN	OTROS
67	Metrics suite for oriented design	CHINDAMBER, Shilam, KEMERER, Chis	http://portal.acm.org/citation.cfm?id=631131	IEEE Transactions on software engineering. Vol 20, N° 6. Junio 1994.
68	Las Métricas de Software y su Uso en la Región	GONZALEZ, Heidi	http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/gonzalez_d_h/indice.html	Escuela de Ingeniería. Departamento de Ingeniería en Sistemas Computacionales. 2001
69	An object-oriented metrics suite for Ada 95	PRITCHETT, William IV	http://portal.acm.org/citation.cfm?id=507546.507604	Portal ACM. 2001
70	A validation of object-oriented design metrics as quality indicators. Technical Report	BASILI, Victor, BRIAND, Lionel, MELO, Wacélio	http://www.computer.org/portal/web/csdl/doi/10.1109/32.544352	Univ. of Maryland, Dep. of Computer Science, College Park, MD, 20742 USA. 1995
71	Object-Oriented Design Quality Models – A Survey and Comparison	EL-WAKIL, Mohamed, EL-BASTAWISI, Ali, BOSHRA, Mokhtar, and FAHMY, Ali	http://homepages.wmich.edu/~m5elwakil/INFOS04_EIWakil.pdf	2nd International Conference on Informatics and Systems, March 2004
72	Object-Oriented Metrics that Predict Maintainability	LI, Wei, HENRY, Sally	Object-Oriented Metrics that Predict Maintainability	Journal of Systems and Software, vol 23 no.2, pp.111-122, 1993
73	Code and design metrics for object-oriented systems	LINDROOS, Jaana	http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.98.2795&rep=rep1&type=pdf	Seminar on Quality Models for Software Engineering. Department of

No.	TITULO	AUTOR (ES)	DISPONIBLE EN	OTROS
				Computer Science, UNIVERSITY OF HELSINKI. 2004
74	Object-Oriented Design Quality Metrics	ANDERSSON, Magnus, VESTERGREN, Patrik	http://www.citerus.se/download/18.a939951052f77843580001347/1100-1836_object_oriented_design_quality_metrics.pdf	Informaton Technlogy, Computing Science Department. Uppsala University. 2004
75	The FAMOOS Object-Oriented Reengineering Handbook	BAR, Holger, et. al	http://www.iam.unibe.ch/~famoos/handbook	
76	Software Measurement and Estimation: A Practical Approach	LAIRD, Linda, BRENNAN, Carol	http://www.amazon.com	Wiley - IEEE Computer Society Press. 2006
77	Software Metrics	FENTON, Norman, PFLEEGER, Lawrence	http://www.amazon.com	PWS Publishing Co. 2ª Edición, 1996
78	Object-oriented metrics for early system characterization: A crc card-based approach. Technical Report TR 94-107	OJHA, Neeraj, MCGREGOR, Jhon	http://www.cs.clemson.edu/~johnmc/papers/METRIC_S/crcmetric.ps	Dept. of Computer Science, Clemson University, 1994
79	Evaluating the Impact of Object-Oriented Design on Software Quality	ABREU, Fernando, MELO, Walcélio	http://portal.acm.org/citation.cfm?id=823874	Proceedings of the 3rd International Symposium on Software Metrics: From Measurement to

No.	TITULO	AUTOR (ES)	DISPONIBLE EN	OTROS
				Empirical Results. (METRICS'96). IEEE, Berlin, Germany, March 1996
80	OO Process and Metrics for Effort Estimation	CHAMPEAUX, Dennis, HORNER, Simon, MILLER, Granville	http://portal.acm.org/citation.cfm?id=260268	ACM Press, vol 6, pg 138-142.1995
81	Una Propuesta Para la Verificación de Requisitos Basada en Métricas	BERNÁRDEZ, B, DURÁN, A, TORO, M	http://www.aemes.org/rpm/descargar.php?volumen=1&numero=2&articulo=2	Revista de Procesos y Métricas de las Tecnologías de la Información (RPM) VOL. 1, N° 2, 2004
82	Object-Oriented Design Heuristics	RIEL, Arthur	www.amazon.com	Addison Wesley. 1ª Edición, 1996
83	Software Formal Inspections Guidebook	NASA	http://satc.gsfc.nasa.gov	
84	DICCIONARIO INGLÉS-ESPAÑOL DE TÉRMINOS DE GESTIÓN DEL SERVICIO DE TECNOLOGÍAS DE LA INFORMACIÓN (TI)	OGC, itSMF INTERNATIONAL, itSMF ESPAÑA Y EXIN	http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/itil_glosario.htm	
85	Caracterización de Productos Software con Métricas no Redundantes	MANSO, Esperanza, CRESPO, Yania, DOLADO José Javier	http://www.giro.infor.uva.es/oldsite/docpub/caracteriza_prod_sw.pdf	JISBD 2002: 177-188
86	Maintainability	LINDELL, Jonas,	http://web.abo.fi/~kaisa/HL.pdf	

No.	TITULO	AUTOR (ES)	DISPONIBLE EN	OTROS
	metrics for object oriented systems”.	HAGGLUND, Mats		
87	Object-Oriented Software Metrics: A Practical Guide	LORENZ, Mark, KIDD, Jeff	http://portal.acm.org/citation.cfm?id=177063	Prentice-Hall. 1994
88	Object-Oriented Metrics in Practice	LANZA, Michele, MARINESCU, Radu	www.amazon.com	Springer. 2006
89	Ingeniería del Software: principios y conceptos	CHOQUE, Guillermo	http://www.institutopascal.edu.ar/Alumnos/AsignaturasAnalistasDeSistemas/1ano/ArquitecturaDelComputador/IngenieriaSoft.pdf	2002
90	A Survey of Metrics for UML Class Diagrams	GENERO, Marcela, PIATTINI, Mario, ALARCOS, Coral	http://www.jot.fm/issues/issue_2005_11/article1/	JOURNAL OF OBJECT TECHNOLOGY. Vol. 4, No. 9, November-December 2005.
91	Software Reliability: Principles and Practices	MYERS, Glendfor	www.amazon.com	New York, NY: John Wiley & Sons; 1ª Edición. 1976
92	A Complexity Measure	McCABE, Thomas	http://www.literateprogramming.com/mccabe.pdf	IEEE Transactions on Software Engineering, December 1976.
93	Software Metrics. SEI Curriculum Module SEI-CM-12-1	MILLS, Everaldo	http://www.sei.cmu.edu/reports/88cm012.pdf	Carnegie Mellon University. 1988
94	Structured Testing: A Testing Methodology Using the Cyclomatic Complexity Metric	McCABE, Thomas, WATSON, Arthur	http://www.mccabe.com/pdf/nist235r.pdf	NIST Special Publication 500-235. 1996
95	Object-Oriented Metrics: an overview	BELLIN David, TYAGI, Manish,	http://portal.acm.org/citation.cfm?id=782185.782189	IBM Centre for Advanced Studies

No.	TITULO	AUTOR (ES)	DISPONIBLE EN	OTROS
		TYLER, Maurice		Conference. 1994
96	A Qualitative Evaluation of a Software Development and Re-Engineering Project	PANAS, Thomas, LINCKE, Rudiger, LUNDBERG, Jonas, LOWE, Welf	http://portal.acm.org/citation.cfm?id=1106686	IEEE Computer Society. 2005
97	Cohesion and reuse in an object-oriented system	BIEMAN, James, KANG, Byung-Kyoo	http://portal.acm.org/citation.cfm?id=223427.211856	ACM press. 1995
98	Implementing Automatic Quality Verification of Requirements with XML and XSLT	DURAN, Amador, RUIZ, Antonio, TORO, Miguel	http://en.scientificcommons.org/4230780	2008
99	Software Testing Guide Book. Part I: Fundamentals of Software Testing	AJITHA et. al	http://www.SofTReL.org	Software Testing Research Lab
100	Empirical Investigations in Software Architecture Completeness	LANGE, Christian	http://www.win.tue.nl/~clange/empanada/papers/The_sis_CLange.pdf	Technische Universiteit Eindhoven. 2003.
101	Latin Bussiness Ltda		http://www.trabajofreelance.com/do/perfil-latinbs	
102	SITI Ltda		http://www.siticol.com	
103	Input Technologies Ltda		www.input.com.co	
104	Seratic Ltda		http://www.seratic.com	

ANEXO V: SEGUIMIENTO DE ACTIVIDADES

En la ejecución del proyecto fue realizada la recopilación y análisis de la información encontrada acerca de modelos y estándares dedicados a la calidad del producto software, además de documentos relacionados que tratan de los temas relevantes para este trabajo como lo son atributos, métricas y heurísticas. Para poder cumplir con cada objetivo fueron realizadas distintas actividades, las cuales arrojaron ciertos resultados que serán citados en este anexo. Con el fin de que el lector sea guiado en el proceso a que tuvo lugar esta investigación es mostrada continuación el trabajo realizado en cada fase:

V1. Fase 1: Delimitación de la población

Para esta fase se hizo uso de la investigación descriptiva, pues permitió conocer por medio de encuestas aspectos importantes del manejo de la calidad en una muestra de las empresas y el sector académico. En esta fase se realizaron una serie de encuestas, en busca de delimitar la población y se obtuvieron los resultados siguientes:

Población: Inicialmente se pensó en la población a la que estaría dirigida esta investigación pensando así en estudiantes de últimos semestres o en realización de trabajo de grado de la Universidad del Cauca, además de empresas productoras de software ubicadas en la región sur occidental de Colombia.

Muestra: Para la muestra se tomó la encuesta a 15 personas entre estudiantes e integrantes de las empresas en la ciudad de Popayán.

Nombre de la encuesta: Encuesta para el diagnóstico respecto a prácticas del producto software

Resultados de la encuesta:

- Falta completitud en la información que los encuestados tienen acerca de la valoración de calidad para el producto software. En algunas ocasiones se conoce la terminología pero no el significado.
- Existe en conocimiento mínimo acerca de las técnicas y herramientas que se pueden utilizar en la valoración de la calidad de un producto software o no se tiene acceso a ellas. Sin embargo aquellos que si hacen uso de estos recursos prefieren las herramientas para la valoración.
- En la encuesta se encontró que varios de los encuestados a pesar de tener conocimiento de calidad de software, no lo aplican de manera adecuada o no lo utilizan.
- La totalidad de los encuestados manifiestan conocer y aplicar el paradigma orientado a objetos
- Se encontró que la totalidad de los encuestados utiliza algún tipo de arquitectura de software en sus proyectos.
- Existe un número mínimo de encuestados que realizan pruebas al producto software que desarrollan.

Para ampliar la información de los resultados de la encuesta en esta fase se puede referir al Anexo S y lo concerniente a las demás encuestas se pueden consultar en el artículo W1 del anexo W

V2. Fase 2: Recopilación de la información

El tipo de investigación que se usó en esta fase fue la histórica, con la cual se realizó la recopilación de la información de los modelos y estándares para calidad de producto software y documentación relacionada, abstrayendo aquellos elementos que fueron de interés para la conformación del marco conceptual.

Para la búsqueda de información se tuvo en cuenta fuentes digitales e impresas. En el caso de las fuentes digitales se tuvo en cuenta las páginas de la ISO, IEEE, Portal ACM y Bibliotecas virtuales de universidades alrededor del mundo. En el caso de documentación impresa, se consideró información de Bibliotecas nacionales y la compra de material a través de páginas como Amazon y Ebay. Después de la búsqueda de información se clasificó como fuentes primarias aquellas en donde se contenía el modelo o estándar de calidad, como por ejemplo “*Factors in software Quality*” de Jim McCall e “*ISO 9126*” de ISO/IEC. Como fuentes secundarias se tuvo en cuenta artículos y libros en donde se mencionan los modelos, atributos, métricas o heurísticas. Para ver los documentos tenidos en cuenta y su fuente de obtención por favor refiérase al anexo U

Ya con la documentación necesaria se procedió a analizar la información que presentaron dichos documentos, llegando a los resultados que se muestran en los anexos G, H e I. Finalmente se realizó un primer informe de lo encontrado en relación a atributos, en donde se omiten las métricas y heurísticas porque hasta ese entonces, aún no se habían terminado de analizar. Éste informe puede ser consultado en el anexo W literal W2.

V3. Fase 3: Análisis de la información y conformación del marco conceptual.

Ya para la última fase se hizo uso de la investigación descriptiva para explicar las relaciones que existían entre los elementos (estas se pueden ver en el capítulo 3 y 4 de la monografía). Además, como resultado del capítulo 4 se presentó el algoritmo de uso del marco como propuesta para poder acceder a la información y usarla.

En este punto se identificaron los elementos necesarios para el marco conceptual, dando como resultado 134 atributos, 617 métricas y 185 heurísticas, las cuales se organizaron en las capas propuestas. Además se propuso un espectro de tres niveles (claro, gris y oscuro) para clasificar los elementos y de esta forma tener una visión más objetiva de cada elemento.

En el momento que se conformó el marco conceptual de atributos, métricas y heurísticas, se realizó una encuesta en donde se buscaba saber qué atributos evaluar, los resultados se pueden ver en el anexo P sección P6; de esta forma se pensó en la aplicación de la evaluación de un sub producto en cada una de las empresas con el fin de verificar los resultados del marco. Los resultados obtenidos después de aplicar las métricas a 4 sub productos software dieron un 80% de aplicación del marco, lo cual indica una proximidad a llegar a empresas con características de MiPyMesPS. El informe se puede ver en ver anexo Q sección Q6

Finalmente, con la realización de las tres fases, se dió por concluido el proyecto y los resultados se pueden observar a lo largo de la monografía y los anexos.

ANEXO X: CARTAS DE CERTIFICACION DE ASISTENCIA A LAS EMRESAS

X1. *Latin Business*



LATIN BUSINESS LTDA

CERFIFICA
A SOLICITUD DEL INTERESADO QUE

Liliam Paola Bolaños y Manuel Alejandro Navia, realizaron visitas a nuestras instalaciones en los meses de marzo y octubre de 2009, donde recolectaron datos para su proyecto de grado; además hicieron entrega la apreciación y respectivas recomendaciones.

Para constancia se firma en Pasto, a los 9 días del mes de Noviembre de 2009

JOHN JAIRO ORTIZ CANCHALA
Gerente (E)

Cra. 43 No. 18A-130 – Teléfono: 7310618 San Juan de Pasto
Cel: 3007867357
latin@parquesoftpasto.com

X2. SITI



San Juan de Pasto, 10 de Noviembre de 2009

SOLUCIONES INTEGRALES EN TECNOLOGÍAS DE LA INFORMACIÓN S.I.T.I. LTDA. NIT. 900.032.188-8

CERTIFICADO DE CUMPLIMIENTO

Se certifica a la Universidad del Cauca, que los estudiantes LILIAM PAOLA BOLAÑOS y MANUEL ALEJANDRO NAVIA, realizaron visitas a nuestras instalaciones en los meses de marzo y octubre de 2009 para aplicar encuestas, evaluaciones y formatos de recolección de información de desarrollo de software para su proyecto de grado.

Durante este tiempo también realizaron recomendaciones frente a nuestros mecanismos de trabajo, nos entregaron información frente a las practicas actuales de desarrollo y mantuvieron una comunicación fluida para apoyar nuestros procesos internos.

Para constancia se firma en Pasto, a los diez días del mes de Noviembre de 2009

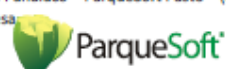
Atentamente,

GELBER ORLANDO MORAN SILVA
Coordinador de Desarrollo
Soluciones Integrales en
Tecnologías de la Información
S.I.T.I. Ltda.
Cel. 315 559 7677

www.siticol.com

www.siticol.com – info@siticol.com – Carrera 43 No. 18ª -130 Pandiaco – ParqueSoft Pasto – (57) 27310618

Somos una empresa



X3. SERATIC



X4. INPUT



INPUTTECHNOLOGIES LTDA.
NIT: 900076583-3

CERFIFICA
A SOLICITUD DEL INTERESADO QUE

Liliam Paola Bolaños y Manuel Alejandro Navia, realizaron visitas a nuestras instalaciones en los meses de febrero y noviembre de 2009, donde recolectaron datos para su proyecto de grado; además hicieron entrega de la apreciación y respectivas recomendaciones.

Para constancia, se firma a los 13 días del mes de Noviembre de 2009.

A handwritten signature in black ink that reads 'Ricardo Andrés Ledezma'.

Ricardo Andrés Ledezma.
Ingeniero de Proyectos
INPUT TECHNOLOGIES LTDA.

Carrera 15 # 5-41 Oficina 205 Popayán
Teléfonos: 8317533 - 3013715782
E-M@IL: info@input.com.co

ANEXO Y: CARTAS DE RECEPCION DE ARTICULOS

Y1. *Revista LASALLISTA de investigación*

Nombre: Revista Lasallista de Investigación.
ISSN: 1794-4449
Institución: Corporación Universitaria Lasallista.
Pais: Colombia.
Categoría: B.



Caldas, 9 de diciembre de 2009

La Directora de Investigación de la Corporación Universitaria Lasallista y Editora
de la Revista LASALLISTA de Investigación

Hace constar que:

Jorge Jair Moreno Chaustre, Liliam Paola Bolaños Rengifo y, Manuel Alejandro Navia Porras, presentaron a consideración del Comité Editorial de la Revista LASALLISTA de Investigación el artículo "*Un acercamiento a las prácticas de calidad de software en las MiPyMESPS del Sur-occidente colombiano*". Durante el primer trimestre de 2010, el Comité dará respuesta con base en el concepto de los evaluadores.

Cordialmente,


MARÍA CONSUELO CASTRILLÓN AGUDELO
Directora de Investigación

Y2. *Revista UIS Ingenierías*

Nombre: Revista UIS Ingenierías.
ISSN: 1657-4583
Institución: Universidad Industrial de Santander.
País: Colombia.
Categoría: C.

Carta de recepción en espera.