

## **ANEXO A**

**Reseña de artículos para estado del arte**

## **ANEXO A.**

### **Reseña de artículos para estado del arte**

En este <nexo se presenta información ms detallada de los artículos obtenidos con las cadenas de búsqueda que se emplearon durante la revisión sistemática. Las cadenas de búsqueda empleada y su denominación se indican a continuación:

**CADENA A:** "Unit testing metrics" **OR** "small software development enterprises"

**CADENA B:** "Integration testing metrics" **OR** "small software development enterprises" **OR** "Methodology, procedure, framework"

## A.1. Reseña de los artículos obtenidos

### Art 01

J. Brzezinski, D. Dwornikowski, M. Kalewski, T. Pawlak y M. Sajkowski. "MDL: Metrics Definition Language", In: *Nguyen N.T., Kim CG., Janiak A. (eds) Intelligent Information and Database Systems. ACIIDS 2011. Lecture Notes in Computer Science*, vol 6591, pp. 248-256. 2011.

- Contexto  
Este artículo nos muestra que la mayoría de sistemas de monitoreo como (SOA) arquitectura orientada a servicios y cloud computing tienen deficiencias para crear y definir métricas dinámicamente a partir de métricas ya existentes. Es por esto que se propone un formato universal para definir métricas, se establecen los requisitos y la gramática de MDL junto con ejemplos del uso de MDL. Se compara el MDL con el lenguaje de procesamiento de eventos (EPL), en el que MDL basa libremente su gramática. Se destacan las diferencias entre estos.
- Problema a resolver  
Las métricas comúnmente usadas y definidas estáticamente ahora pueden no ser suficientes para describir completamente los sistemas distribuidos complejos y dinámicos.
- Validación  
Se propone un lenguaje de definición de métricas el cual proporciona la base suficiente para crear y definir métricas adecuadas para sistema. Se dan los requisitos y la gramática.

Requisitos para MDL:

- ❖ Legibilidad humana y mecánica: debe ser fácil de entender y usar por un humano, pero también fácil de procesar por una máquina
- ❖ Universalidad: debería ser bastante fácil de aplicar en la mayoría de los sistemas de monitoreo existentes

- ❖ Separación clara de fuentes de datos y métricas: debe proporcionar la sintaxis para las definiciones de origen de datos por separado de las definiciones de métricas.
  
- ❖ Soporte para métricas complejas: debería proporcionar formas para una definición fácil y flexible de nuevas métricas.
  
- ❖ MDL debe ser independiente del formato de datos: Este requisito supone que nuestro idioma no debe asignarse estrictamente a un formato específico del evento o paquete de datos.
  - Qué le falta  
En el futuro planeamos implementar métodos para determinar el estado global consistente en sistemas distribuidos para la verificación de métricas globales. Nos gustaría extenderlo para admitir más escenarios de monitoreo e integrar presumiblemente su nomenclatura con estándares populares, como CIM o WSDM.

## Art 02

M. English Lero, T. Cahill y J. Buckley. "Construct specific coupling measurement for C++ software". En: *Journal Computer Languages, Systems and Structures*, vol. 38 issue 4, December, 2012, pp. 300-319.

- Contexto  
Se hace un recuento de las métricas que existen para medir el acoplamiento y se propone un conjunto de métricas de software que miden la cantidad de acoplamiento habilitado por diferentes construcciones de lenguaje de programación en C++ (como amistad y herencia). Se da un modelo formal de las estructuras en c++.  
Se proponen métricas para saber las interacciones entre las clases-hijos al igual que clase-antecesoros y otra para determinar el número de métodos en una clase que acceden a miembros ocultos que la declaran como amiga.
- Problema a resolver  
Muchas construcciones de estructuración del sistema en lenguajes OO se han diseñado para ayudar a minimizar el acoplamiento entre módulos y maximizar la cohesión y la encapsulación de los módulos. el acoplamiento se centra en las interacciones entre clases y la cohesión se centra en las interacciones internas de las clases. Sin embargo, no está claro si el sw desarrollado utilizando estos lenguajes usa las construcciones disponibles de manera adecuada para garantizar que se adhiera a estos principios
- Validación  
Este artículo aborda los efectos específicos sobre el acoplamiento de la herencia, la amistad y la visibilidad de los miembros, en sistemas escritos en C++  
El valor central de este trabajo radica en su desambiguación de posibles mecanismos de acoplamiento que se superponen en el software C ++. Con esta desambiguación, tiene una mayor comprensión del uso de estos mecanismos.  
Se presenta un modelo de datos formal para facilitar la definición formal de las métricas.

- Que le falta  
Les faltó la implementación para grandes sistemas de sw porque sería demasiado lenta con estas métricas, haría falta una automatización para mejorar el tiempo de análisis

### Art 03

V. Kumar, A. Sharma, R. Kumar y P.S. Grover. "Quality aspects for component-based systems: A metrics based approach". En: *Journal of Software: Practice and Experience*, vol. 42, issue 12, December 2012, pp. 1531-1548.

- Contexto  
El presente trabajo produce análisis crítico, características de calidad, complejidad, dependencia, reutilización, mantenibilidad para componentes y sistemas basados en componentes (CBS).
- Problema a resolver  
Se quiere resolver la complejidad que existe para la medición de calidad en sw basado en componentes
- Validación  
Se centraron en la literatura de investigación, que coincidió con el objetivo como el enfoque basado en métricas para la calidad de la CBSE. Se aplicó una búsqueda sistemática para encontrar tanta literatura como sea posible, y se encontraron adecuados 49 artículos, que se analizan en función de parámetros definidos.

QUALITY ASPECTS FOR COMPONENT-BASED SYSTEMS: A METRICS BASED APPROACH

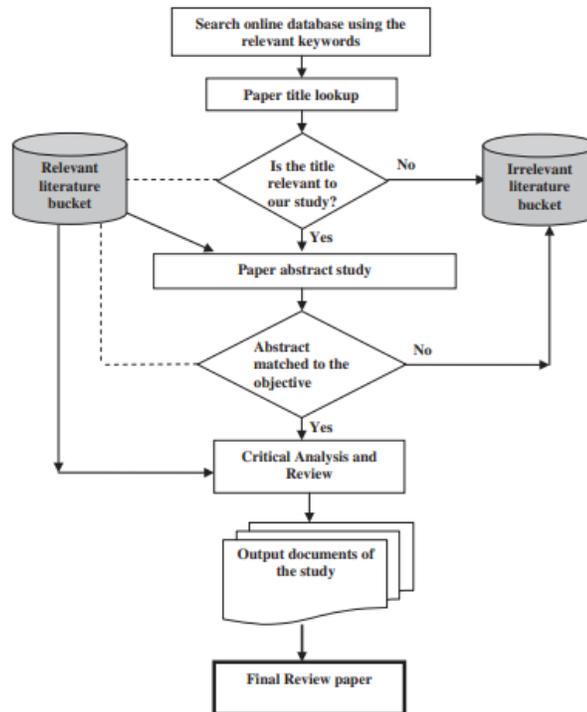


Figure 1. The review process

- Que le falta
  - Se pueden hacer modelos más robustos y validados en esta área. Se necesita una mayor exploración de los enfoques basados en métricas para la calidad CBSE, que pueden atender a los siguientes parámetros principales en el modelo / métrica:
    - ❖ Debería necesitar menos datos para la precisión
    - ❖ Los datos de entrada deben ser recopilables con menos esfuerzo.
    - ❖ Los métodos completamente validados matemáticamente
    - ❖ Enfoques más prácticos

## Art 04

S. Stevanetic y U. Zdun. "Software Metrics for Measuring the Understandability of Architectural Structures: a systematic mapping study". En: *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering*, p.1-14, April 27-29, 2015, Nanjing, China [doi>10.1145/2745802.2745822].

- Contexto

se reconoce que la arquitectura de software juega un papel clave en la gestión y el mantenimiento del sistema de software, en general una descripción arquitectónica puede comprender múltiples vistas que describen el sistema concentrándose en una de las muchas preocupaciones del sistema, como las vistas lógicas, de implementación, implementación o proceso, y desde el punto de vista de diferentes partes interesadas, tales como usuarios finales, desarrolladores, gerentes de proyectos y analistas de negocios

este estudio se enfoca en la información arquitectónica capturada en las estructuras de nivel superior con respecto a sus relaciones con la implementación del sistema. este estudio es más detallado porque se ve concretamente las métricas a nivel de arquitectura desde la perspectiva de implementación del sistema con respecto a los conceptos de comprensibilidad que son importantes en ese contexto

- Problema a resolver

Se plantean ciertas preguntas a resolver en el artículo

¿Qué métricas de software que miden los conceptos comprensibles de las estructuras de software arquitectónico de nivel superior?

¿Cuál es el nivel de madurez de las métricas identificadas?

¿Qué limitaciones en los estudios actuales deberían abordarse en futuras investigaciones? El propósito de esta pregunta de investigación es identificar cualquier deficiencia en los estudios existentes y proponer posibles técnicas y trabajos futuros para mitigarlos.

- Validación

Se clasifican todos los estudios con respecto al modelo de representación que se utilizan para describir la metodología y el contexto de las métricas, y

para definir las métricas que se proponen. Durante el análisis de datos se identifican 3 modelos diferentes que corresponden a tres tipos de enfoques diferentes:

- Métricas basadas en estructuras internas: los estudios clasificados en esta categoría proponen métricas basadas en la estructura interna de artefactos de nivel superior medidos (componentes, módulos, paquetes, etc.) y sus relaciones.
  - Métricas basadas en gráficos: los estudios clasificados en esta categoría proponen métricas basadas en un modelo de gráfico del sistema dado. Los gráficos representan artefactos del sistema y sus interacciones como un conjunto de nodos y aristas. El enfoque basado en gráficos es ampliamente utilizado en ingeniería de software debido a su simplicidad inherente.
  - Métricas basadas en modelos específicos: los estudios clasificados en esta categoría proponen métricas basadas en alguna representación o modelo específico de la estructura del sistema que se requiere para definir las métricas que proponen
- Que le falta  
En general, hay una falta de madurez en los estudios. Se refleja principalmente a través de ambigüedades en las definiciones de métricas, falta de validación empírica y falta de información, como el mapeo de las características de calidad que pretenden medir y la falta de análisis comparativo, lo que oscurece la utilidad de las métricas para los profesionales.

## Art 05

O. Ozgur Tanriover y R. Eryigit. "An empirical analysis of early object oriented design metrics in relation to code size". En: *2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS) Proceedings*, pp. 62 - 65. ISBN: 978-1-4799-8353-7.

- Contexto

Se analizan empíricamente las métricas de OO en relación con el tamaño del software, especialmente las métricas disponibles en la fase inicial de diseño. se proporciona una breve revisión de estudios recientes donde se utilizan métricas para los atributos del software OO. Las métricas de diseño orientadas a objetos son mediciones del estado estático del diseño y también se utilizan para evaluar atributos de calidad, complejidad y predecir clases y componentes potencialmente propensos a fallas por adelantado como indicadores de calidad. Las métricas de diseño también ayudan a identificar posibles problemas en las primeras etapas del proceso de desarrollo

Al final se resumen los principales hallazgos y el posible trabajo futuro basado en los datos recopilados

Como estas métricas pueden generarse automáticamente mediante herramientas de casos, se puede usar un modelo de regresión para estimar las líneas de código del software desde el diseño inicial

- Problema a solucionar

La falta de estudios que permitan generar automáticamente mediante herramientas de casos especialmente las métricas disponibles en la fase inicial de diseño, Solo unos pocos estudios han tenido en cuenta la posible relación del tamaño del software y las métricas de OO

- Validación

tres conjuntos de métricas de nivel de paquete (Martin, MOOD y CK) se evalúan y comparan empíricamente para predecir el número de fallas previas

al lanzamiento y el número de fallas posteriores al lanzamiento en los paquetes. Eclipse, uno de los sistemas de código abierto más grandes, se utiliza como caso de estudio. Los resultados indican que los modelos de predicción que se basan en la suite Martin son más precisos que los que se basan en las suites MOOD y CK en las versiones de Eclipse. El problema de inconsistencia en las métricas en diferentes suites se atribuye a la falta de formalismo en su definición y se presenta una formalización basada en Z de las métricas CK.

- Que le falta

En general, los modelos de predicción de otros atributos de software como la mantenibilidad y la reutilización se pueden desarrollar en base a los datos utilizados en este estudio. Además, el cambio entre diferentes versiones se puede utilizar para impulsar aún más el análisis y la confiabilidad para la predicción de atributos de calidad de software en general

## Art 06

M.O. Cinneide, I. H. Moghadam, M. Harman, S. Counsell y L. Tratt. "An experimental search-based approach to cohesion metric evaluation". En: *Empirical Software Engineering*, February 2017, vol. 22, issue 1, pp 292–329.

- Contexto

no podemos confiar en un conjunto de métricas para evaluar las propiedades del software si no podemos determinar en qué medida están de acuerdo, ni tenemos alguna forma de determinar un probable desacuerdo en el peor de los casos

En este trabajo se pretende observar el comportamiento de las métricas en un entorno práctico. En lugar de verlos simplemente como medidas pasivas para extraer del código fuente, el objetivo es promoverlos al nivel de objetos activos que tengan una opinión sobre la calidad del software. Luego se quiere ver dónde estas métricas entran en conflicto entre sí, y explorar estas áreas de conflicto para comprender mejor por qué las métricas entran en conflicto. Se aborda este problema mediante la introducción de una técnica experimental sistemática para evaluar las métricas de software. se aplica la refactorización automática a un programa, midiendo repetidamente los valores de varias métricas antes y después de aplicar cada refactorización. De esta forma se pueden hacer observaciones empíricas sobre las relaciones entre las métricas

Se revela que métricas aparentemente similares pueden estar en conflicto entre sí, y pueden identificar la fuente del conflicto, proporcionando así una nueva visión de las diferencias entre las métricas.

- Problema a solucionar

No es sorprendente que varios estudios hayan intentado comparar métricas de software para comprender mejor sus similitudes y diferencias.

¿cómo se comparan entre sí los resultados de las métricas que pretenden medir la misma propiedad de software?

¿Pueden las métricas que miden la misma propiedad estar en desacuerdo y con qué fuerza pueden estar en desacuerdo?

- Validación

Una crítica al uso de métricas de software es que a menudo no pueden medir lo que pretenden medir. Esto ha llevado a una proliferación de métricas de software muchas de las cuales intentan medir los mismos aspectos del código.

En este artículo se aborda este problema mediante la introducción de una técnica experimental sistemática para evaluar las métricas de software. Este enfoque aplica la refactorización automática a un programa, midiendo repetidamente los valores de varias métricas antes y después de aplicar cada refactorización. De esta forma se puede hacer observaciones empíricas sobre las relaciones entre las métricas. Cuando un par de métricas no están de acuerdo con el cambio provocado por una refactorización, se examina las causas del conflicto para obtener una visión adicional de las diferencias entre las métricas.

- Que le falta

El trabajo futuro en esta área implica una serie de temas. En primer lugar, nos hemos centrado en las métricas de cohesión y su interacción. Sería interesante examinar las compensaciones entre los cambios en estas métricas y, por ejemplo, aquellos que intentan medir el acoplamiento, particularmente porque el bajo acoplamiento y la alta cohesión son características deseables del sistema y podrían considerarse objetivos conjuntos de los diseñadores de sistemas y mantenedores. Del mismo modo, sería interesante estudiar cómo la cohesión semántica (medida, por ejemplo, con WordNet) se compara con la cohesión estructural en este tipo de experimentación. En segundo lugar, nuestro enfoque experimental basado en la refactorización puede desarrollarse aún más para crear una herramienta práctica y fácil de usar que permita a los investigadores y profesionales de las métricas evaluar experimentalmente métricas nuevas o existentes. En tercer lugar, si bien nuestro trabajo ha resaltado el conflicto métrico, no hemos intentado resolver este conflicto y determinar qué métrica es la más adecuada bajo qué circunstancias. El trabajo adicional para abordar este problema deberá incluir estudios que generen la opinión del desarrollador. Finalmente, hemos utilizado sistemas de código abierto en este estudio; Sería valioso aplicar el mismo análisis a sistemas propietarios y comparar los resultados obtenidos en cada caso.

## Art 07

M. Alshayeb, Y. Shaaban y J. Al-Ghamdi. "SPMDL Software Product Metrics Definition Language". En: *ACM Journal of Data and Information Quality*, Vol. 9, No. 4, Article 20. 2018.

- Contexto

En este artículo, se propone el Lenguaje de definición de métricas de productos de software (SPMDL). Se desarrolla un lenguaje de descripción basado en XML para definir métricas de software de forma precisa y reutilizable.

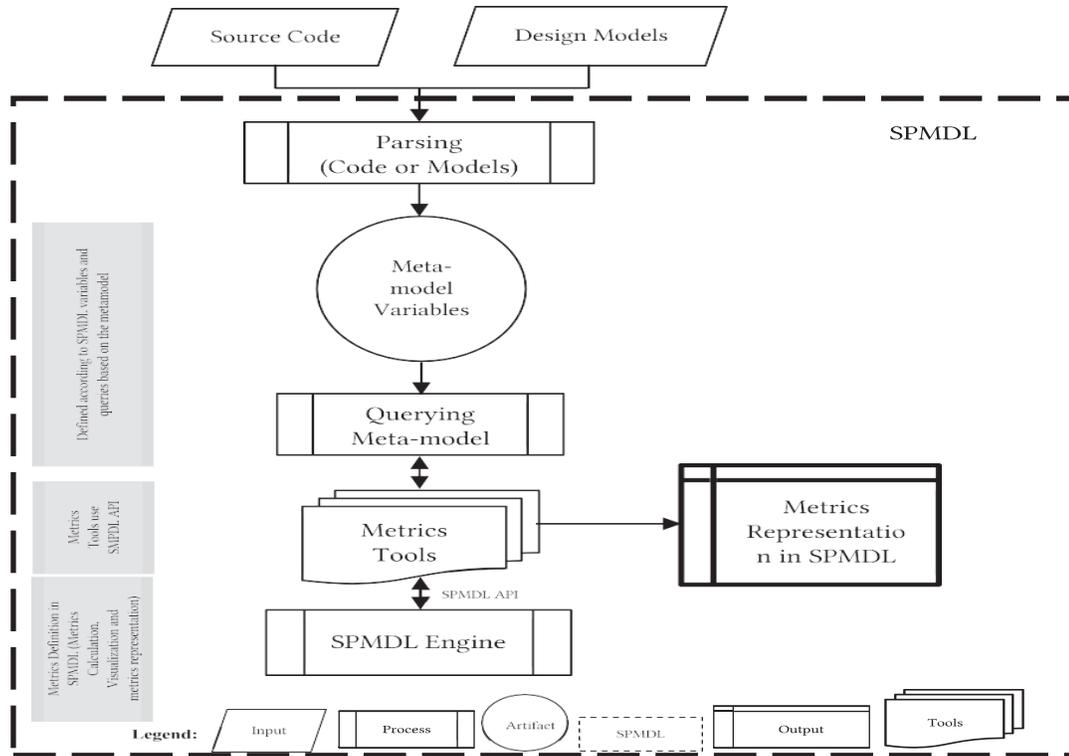
La falta de un formato adecuado para presentar definiciones de métricas de software hace que sea difícil acomodar nuevas métricas en las herramientas existentes de una manera consistente que garantice la compatibilidad y la precisión de la medición.

- Problema a solucionar

no existe una forma estándar para representar definiciones de métricas, lo que sería útil para el intercambio y la personalización de métricas.

La falta de un formato adecuado para presentar definiciones de métricas de software

- Validación



- Que le falta

El trabajo futuro puede incluir análisis de datos, interpretación y clasificación de métricas. También puede incluir computación avanzada como similitud, estabilidad y el soporte para la visualización al proporcionar una descripción de visualización de la definición de métrica para admitir jerarquías de visualización de métrica comunes, también Proporcionar una versión de SPMDL que se ejecute en la nube.

## Art 08

D. Mourtzis, S. Fotia y N. Boli. "Metrics definition for the Product-Service System Complexity within Mass Customization and Industry 4.0 Environment". En: *2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC) Proceedings*, pp. 1166-1172. ISBN: 978-1-5386-0774-9. 2017.

- Contexto

En este artículo se propone una metodología para medir la complejidad de personalización de sistemas de productos y servicios en el entorno de la industria 4.0 (la industria 4.0 consiste en la digitalización de la industria y todos los servicios relacionados con la empresa). Se introducen métricas para la complejidad de PSS buscando una fácil cuantificación de la complejidad de los sistemas en función del número de componentes.

- Problema a solucionar

Pretende mejorar la evaluación de PSS ya que en la actualidad es un campo inmaduro, con pocos resultados.

- Validación

La validación se hace al proponer una metodología para medir la complejidad de personalización de PSS en el entorno de la Industria 4.0.

La metodología se basa en el concepto de ElMaraghy et al. (Fig. 1), según el cual, tres son los factores principales que afectan la complejidad de un sistema: i) la cantidad de información que un sistema tiene que gestionar, ii) la diversidad / variantes de la información de intercambio, y iii) el contenido de los mensajes que transfieren la información.

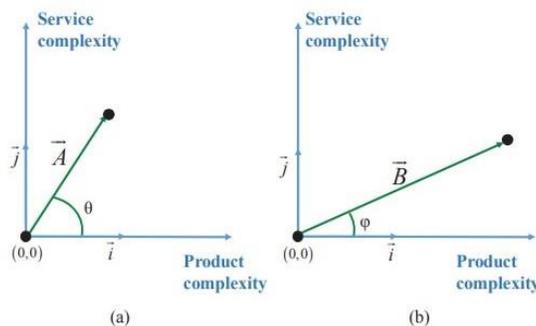


Fig. 2. Product-Service complexity due to: (a) the quantity of the product

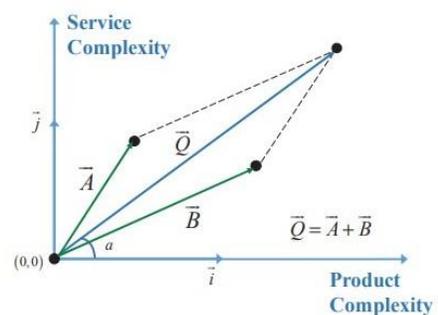


Fig. 3. Complexity due to quantity and variants

- **Que le falta**  
El trabajo futuro incluirá la mejora de la metodología considerando más aspectos de las tecnologías emergentes de la Industria 4.0, así como la aplicación de la metodología propuesta en casos industriales de la vida real. Además, las métricas de complejidad de PSS se formularán de una manera más genérica, incluidas otras variables relacionadas con el PSS, y los pesos relevantes con el objetivo de abordar las necesidades / naturaleza de las diferentes empresas industriales.

## Art 09

E. Tempero y P. Ralph. "A Framework for Defining Coupling Metrics". En: Science of Computer Programming, issue 166, pp. 214–230, 2018.

- Contexto  
“grado de interconexión de las partes de un sistema” la relación entre acoplamiento y diseño es lo que motiva a los investigadores y profesionales a desarrollar métricas específicamente para medir este fenómeno que se da la mayor parte en diseños orientados a objetos, se han propuesto muchas, es por eso que se dan las siguientes preguntas: ¿todas las métricas miden lo mismo? El ejemplo que se cita es el de los grados de temperatura Celsius y Fahrenheit, son métricas muy diferentes, pero al final miden lo mismo. Se habla de un concepto métricas subespecificadas, por ejemplo, si al implementar métricas para el código que refiere a bibliotecas no quiere decir que se vaya a incluir el análisis de bibliotecas, entonces se da la misma métrica de diferentes maneras por lo tanto dará diferentes resultados, es a lo que se llama metaanálisis de acoplamiento, el objetivo principal es investigar la medida en que dos o más métricas de acoplamiento miden lo mismo. El resto del artículo se da entre definiciones y en una parte discute como el marco formal de acoplamiento puede implementarse para introducir una herramienta que mida métricas de pling para código java, se hace la evaluación y por último se dan las conclusiones
- Problema a resolver  
Como proporcionar definiciones inequívocas de métricas de acoplamiento, comparar de manera convincente y confiable las métricas de acoplamiento, como implementar eficientemente las métricas existentes
- Validación  
Propone un modelo de acoplamiento, donde describe el marco en términos de modelo que hace posible trabajar y muestra cómo se puede definir y comparar con las métricas existentes. Para resumir, se pretende lo siguiente: definir, comparar, resolver ambigüedades e implementar. Este estudio se centra en definir las métricas de acoplamiento existentes, sin embargo luego de la definición no se tiene en específico la dependencia entre las entidades a medir, el beneficio de este modelo es simple y aun así ha sido posible proporcionar la definición de algunas métricas.

- Que le falta  
No todas las métricas propuestas o existentes sin acoplamiento se pueden definir en este modelo, es por eso que este estudio abre la puerta para trabajar en este sentido.

## Art 10

A-J. Molnar, A. Neamtu y S. Motogna. "Longitudinal Evaluation of SW Quality Metrics in Open-Source Applications". En: *Proceeding ENASE 2019 Proceedings of the 14th International Conference on Evaluation of Novel Approaches to Software Engineering*, pp. 80-91. 2019. ISBN: 978-989-758-375-9.

- Contexto  
"una métrica general en el modelo de calidad de software no existe" el artículo propone una evaluación longitudinal de las métricas y sus relaciones en el contexto, esto en tres aplicaciones de código abierto. en resumen, tenemos que, la tecnología crece cada día y con ello crece el uso de aplicaciones que a su vez deben ser cada vez más rápidas, se requiere de nuevas metodologías y herramientas para asegurar que el sistema a desarrollar conserve las mismas características de calidad asegurando que "no puedes controlar lo que no puedes medir" lo que se pretende con este trabajo es dar un fundamento que pueda servir para estudiar la relación entre valores métricos de software, calidad de producto y atributos, las tres aplicaciones son: jEdit, Mente libre, TuxGuitar
- Problema a resolver  
No evidencio ningún problema a resolver, pero el objetivo del estudio es brindar una evaluación métrica basada en las características de calidad de software
- Validación  
Para esto, se hizo:
  1. Una evaluación cuantitativa de las métricas para todas las versiones de las tres aplicaciones de código abierto
  2. Un estudio de exploración a largo plazo (18 años) sobre la evolución de las métricas asociados con calidad de software
  3. Identificación del método estadísticamente correlacionado, se evaluó la correlación entre el contexto de cada aplicación y las métricas pendientes que se dan con el aumento del código
  4. Una comparación de los resultados de este estudio y otros realizados previamente donde emplearon metodologías y herramientas similares

- Que le falta  
Ampliar la investigación a otras aplicaciones (no dice cuales) incluyendo dispositivos móviles y aplicaciones donde el código de interfaz de usuario (GUI) no sea dominante, también indicar si la aplicación de métricas implican buenas prácticas de diseño y desarrollo, y establecer un modelo basado en métricas para la calidad de software

## **ANEXO B**

### **Documento compacto de la Guía Técnica**

# Guía Técnica para La medición y valoración en Actividades de Testing para pequeñas organizaciones Desarrolladoras de Software

## ESTRUCTURA GENERAL DE LA GUIA

Se ha elaborado una guía dirigida a pequeñas empresas que permita incorporar buenas prácticas en medición para dar soporte a las necesidades de información sobre testing de software. La vista general de la guía se muestra en la Figura 1 y sus componentes se describen en la Tabla 1.

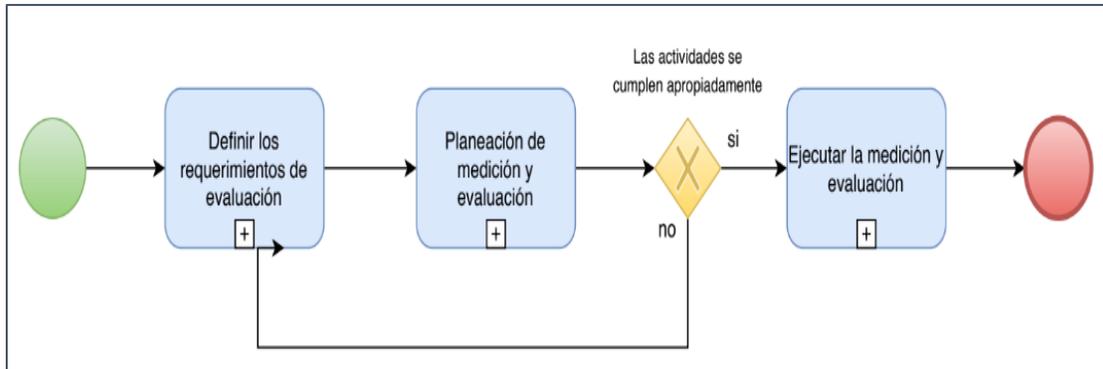


Figure 1. Vista general de la guía técnica

<b>Componentes</b>
<b>Repositorio de métricas.</b> En este incluye las fichas técnicas de todas y cada una de las métricas debidamente estudiadas y seleccionadas, primero se encuentran las métricas directas, luego las métricas indirectas
<b>Ejemplo de uso.</b> Detalla los procesos y las actividades a ejecutar, las entradas y las salidas de cada una de ellas. Contiene los cálculos matemáticos aplicables a las métricas seleccionadas por la organización para valorar los atributos de eficiencia y efectividad, además ofrece tablas de categorización mostrar indicadores referentes a los resultados
<b>Tabla de evaluación de proceso.</b> En esta tabla se registra el reporte resultante de la relación entre los sub-atributos y las entidades seleccionadas

Tabla 1. Descripción de componentes de la guía

## Descripción del Proceso 1 (P1). Definir requerimientos de evaluación

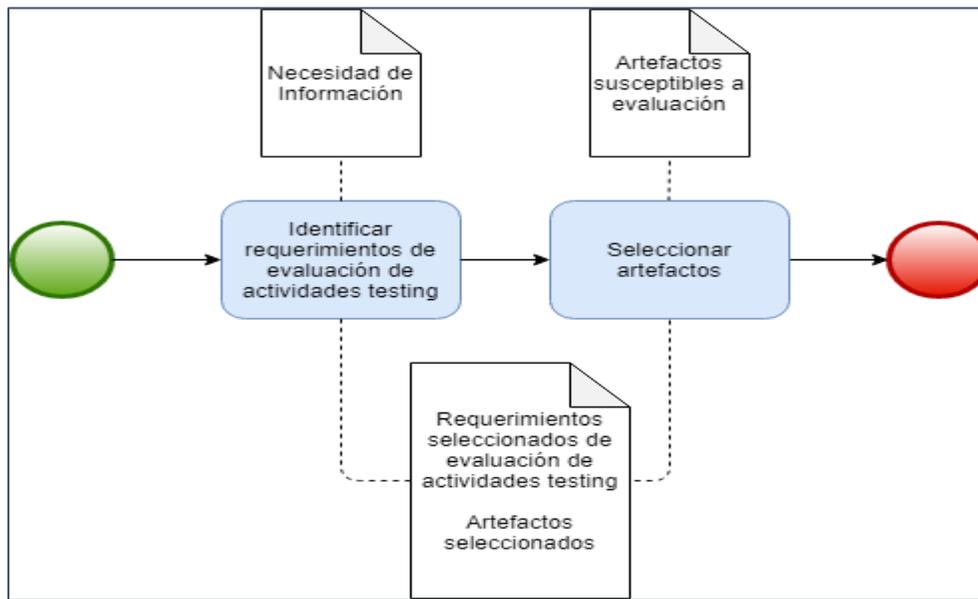


Figure 2. Vista de las Actividades del proceso P1

Entrada(s)	Actividad	Salida(s)
Necesidades de Información	P1.A1. Identificar Requerimientos de evaluación de Actividades de Testing	Requerimientos seleccionados de evaluación de Actividades de Testing
Entidades susceptibles de evaluación	P1.A2. Seleccionar Entidades	Entidades seleccionadas

Tabla 2. Actividades del proceso P1

DESCRIPCIÓN
La primera actividad “Identificar Requerimientos de evaluación de Actividades de Testing” produce una lista de Requerimientos seleccionados de evaluación de Actividades de Testing a partir de diversas fuentes como son las metas de negocio o metas de proyecto.
La segunda actividad “Seleccionar Entidades” genera una lista de filtrada de las entidades asociadas a las actividades de testing, a partir de una lista de entidades posibles.

## Descripción del proceso 2 (P2). Planear medición y evaluación

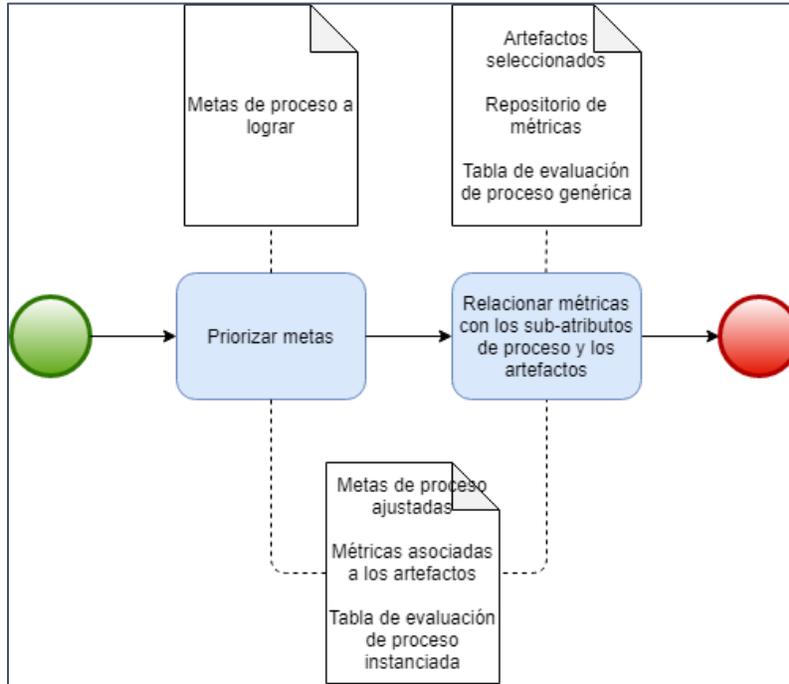


Figure 3. Vista de las actividades del proceso P2

Entrada(s)	Actividad	Salida(s)
Metas de Proceso a lograr	P2. A1. Priorizar metas	Metas de Proceso ajustadas
Entidades seleccionadas Repositorio de Métricas Tabla de Evaluación de Proceso genérica	P2. A2. Relacionar Métricas con los Atributos de Proceso y las Entidades	Métricas asociadas a las Entidades Tabla de Evaluación de Proceso instanciada

Tabla 3. Actividades del proceso P2

DESCRIPCIÓN
La primera actividad “Priorizar metas” usa la lista de metas del proceso y se determina cuáles sub-atributos son de interés para el proyecto en el momento requerido, de modo que se obtenga una lista de metas de proceso ajustadas al momento de avance del proyecto.
La segunda actividad “Relacionar Métricas con los Atributos de Proceso y las Entidades” toma la lista de entidades seleccionados, el Repositorio de Métricas de la presente Guía junto con la Tabla de Evaluación de Proceso en su forma genérica. Esta actividad busca establecer relaciones entre los sub-atributos y las entidades seleccionadas; las cuales se registran en la Tabla de Evaluación de Proceso, de modo que se obtenga una versión instanciada (específica).

### Descripción del proceso 3 (P3). Ejecutar medición y evaluación

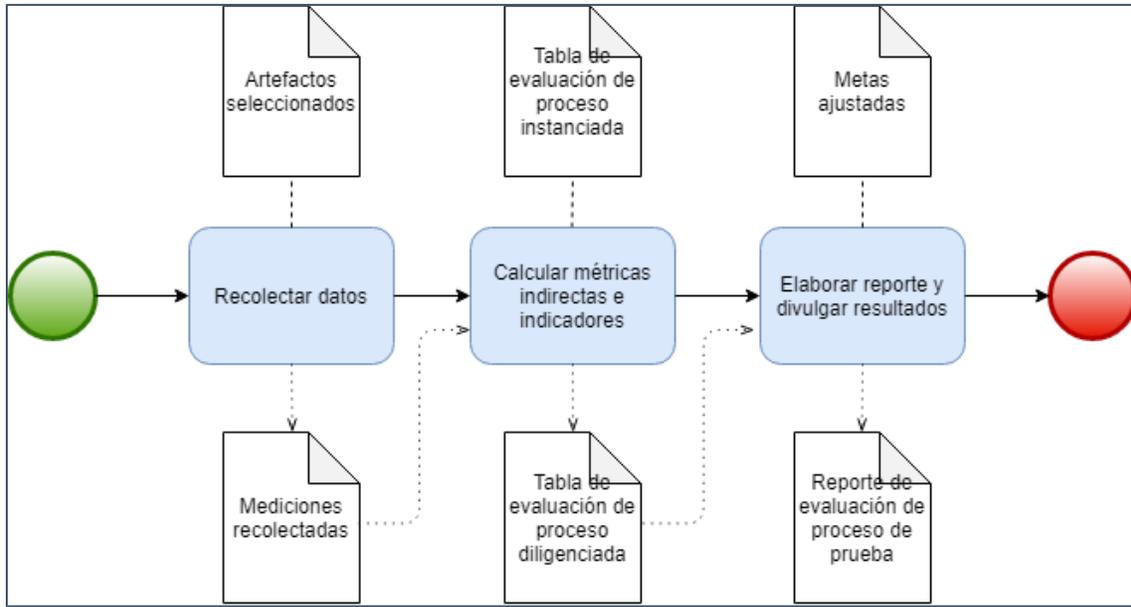


Figure 4. Vista de las actividades del proceso P3

Entrada(s)	Actividad	Salida(s)
Entidades seleccionadas	P3.A1. Recolectar Datos	Mediciones recolectadas
Mediciones recolectadas Tabla de Evaluación de Proceso instanciada	P3.A2. Calcular Métricas Indirectas e Indicadores	Tabla de Evaluación de Proceso diligenciada
Metas ajustadas Tabla de Evaluación de Proceso diligenciada	P3.A3. Elaborar reporte y divulgar resultados	Reporte de evaluación de Proceso de Prueba

Tabla 4. Actividades del proceso P3

<b>DESCRIPCIÓN</b>
La primera actividad “Recolectar Datos” se ha establecido para gestionar la recolección y validación de los datos resultantes de las mediciones directas aplicadas a las entidades seleccionadas.
La segunda actividad “Calcular Métricas Indirectas e Indicadores”, parte de las mediciones recolectadas para obtener los resultados de calcular las métricas indirectas elegidas del repositorio, así como los indicadores que la organización establezca. Los valores se normalizan de acuerdo con lo definido en las fichas técnicas del repositorio de métricas y se registran en la Tabla de Evaluación instanciada para así obtener juicios de valor sobre el estado de los sub-atributos asociados a las actividades de Testing que sean de interés en el momento de aplicación de esta Guía Técnica.
La tercera actividad “Elaborar reporte y divulgar resultados” resume el estado de las actividades de Testing de modo que se puedan identificar sus fortalezas y aspectos por mejorar, el cual se refleja en el reporte de evaluación

## **ANEXO C**

**Documento compacto del ejemplo de uso**

## Resumen del ejemplo de uso para la guía

A continuación, se realizará un escenario para la ejecución de nuestra guía y se mostrará la manera correcta de uso.

Una organización necesita un producto software para el cual ha contratado un equipo de desarrollo para su ejecución, este equipo en aras de conocer la calidad de sus actividades de testing necesita medir los atributos de efectividad y eficiencia de las pruebas relacionadas con las actividades de testing, para esto lleva a cabo la realización de las actividades relacionadas a los procesos que hacen parte de la guía técnica propuesta.

Esta guía se puede utilizar en cualquier momento del desarrollo del producto software, para este ejemplo se tomará para una iteración en un tiempo de una semana de trabajo, los datos se obtendrán a través de la realización de pruebas unitarias.

Del conjunto de pruebas unitarias realizadas en el período de análisis se han recolectado los siguientes datos:

Tabla. Reconocimiento de Métricas utilizadas en el ejemplo de uso

	Ítem	Valor	Código Métrica
1	Cantidad de casos de prueba planeados	15	DIR-012
2	Cantidad de casos de prueba diseñados	12	DIR-028
3	Tiempo invertido en planeación de pruebas	25 horas	DIR-019
4	Tiempo invertido en diseño de pruebas	45 horas	DIR-020
5	Requerimientos aprobados mapeados a casos de prueba	15	DIR-029
6	Total de requerimientos aprobados	20	DIR-030
7	Costo total casos de prueba planeados	\$1.500.000	DIR-031
8	Cantidad de errores encontrados	37	DIR-005
9	Cantidad total de errores encontrados por el equipo de desarrollo	61	DIR-010
10	Cantidad de casos de prueba ejecutados	12	DIR-011
11	Número de defectos corregidos	50	DIR-014
12	Cantidad de casos de prueba aprobados	10	DIR-015
13	Cantidad de casos de prueba bloqueados	2	DIR-016
14	Cantidad de casos de prueba corregidos	10	DIR-017
15	Cantidad de casos de prueba fallados	1	DIR-018
16	Costo de los casos de prueba ejecutados	\$3.600.000	DIR-032
17	Tiempo Total en ejecución de pruebas	60 horas	DIR-001
18	Horas totales empleadas en la corrección de defectos	25 horas	DIR-013
19	Cantidad de defectos corregidos por el equipo de desarrollo	40	DIR-006
20	Cantidad total de defectos en el momento de la medición	61	DIR-007
21	Tiempo total empleado en los casos de prueba	60	DIR-033

### Evaluación atributo EFECTIVIDAD

Se procede a evaluar el atributo EFECTIVIDAD del proceso asociado a un conjunto de pruebas unitarias. Para ello es necesario evaluar los sub-atributos de Efectividad en Preparación de Pruebas (EPP) y Efectividad en Ejecución de Pruebas (EEP) aplicando la expresión:

$$EFECTIVIDAD = (C1 * EPP) + (C2 * EEP)$$

$C1 + C2 = 1$ . Son las ponderaciones para cada sub-atributo. Los valores de las ponderaciones irán de acuerdo a los objetivos de cada empresa y dicho valor puede variar.

De forma genérica, el sub-atributo "Efectividad en Preparación de Pruebas (EPP)" se evalúa así:

$$EPP = (K1 * M1) + (K2 * M2) + \dots + (Kn * Mn)$$

Y el sub-atributo "Efectividad en Ejecución de Pruebas (EEP)" se evalúa así:

$$EEP = (K1 * M1) + (K2 * M2) + \dots + (Kn * Mn)$$

Donde:  $K1 + K2 + \dots + Kn = 1$  son las ponderaciones, pues algunas métricas pueden incidir o "pesar" más para valorar el sub-atributo. [eso lo define la organización y en principio podrían valer igual]

$M1, M2, \dots, Mn$  son los valores normalizados (en escala 0 ... 1 y con sentido positivo) de las métricas que permiten valorar el sub-atributo.

A partir de los datos recolectados, es necesario normalizar las métricas para llevar sus valores a una misma escala de 0..1 con sentido positivo, es decir, que valores altos son los que indiquen un estado deseable.

### Evaluación del sub-atributo de Efectividad en Preparación de Pruebas (EPP).

A partir de los ítems 1 y 3, se calcula la métrica indirecta:

IND-025. Métrica Indirecta "Tiempo medio invertido en planeación de pruebas"

$$TMPP = \frac{TP}{CCP} \quad \text{el resultado es: } TMPP = 15 \text{ horas} / 12 \text{ casos} = 1.25 \text{ horas/caso.}$$

La organización ha definido la siguiente tabla de normalización:

Tabla. Normalización efectividad 1 ejemplo de uso

Rango de valor para la variable x que representa el valor de métrica	Valoración cualitativa	Valor Normalizado
Rango 1: $x \leq 0,4$	Excelente	1
Rango 2: $0,4 < x \leq 0,8$	Aceptable	0.75
Rango 3: valor $0,8 < x \leq 1$	Deficiente	0.4
Rango 4: $x > 1$	Inaceptable	0

De acuerdo con la tabla de normalización, este resultado es inaceptable y su valor normalizado es cero. Esta será la Métrica normalizada 1 y se le ha asignado una ponderación de 0.3.

A partir de los ítems 5 y 6, se calcula la métrica indirecta:

IND-015. Porcentaje de Requerimientos aprobados mapeados a casos de prueba.

$$PRM = \frac{RMCP}{TRA} \quad \text{Cuyo resultado es: } PRM = 15 / 20 = 0.75$$

Esta métrica ya se encuentra normalizada en su definición. Esta será la Métrica normalizada 3 y se le ha asignado una ponderación de 0.4.

**Valoración del sub-atributo:**

$$EPP = (K1 * M1) + (K2 * M2) + \dots + (Kn * Mn)$$

$$EPP = (0.3 * 0) + (0.3 * 0) + (0.4 * 0.75) = 0 + 0 + 0.3 = 0.3$$

**Evaluación del sub-atributo de Efectividad en Ejecución de Pruebas (EEP).**

De igual manera se realiza la valoración del sub-atributo de efectividad en ejecución de pruebas

$$EEP = (K1 * M1) + (K2 * M2) + \dots + (Kn * Mn)$$

$$EEP = (0.4 * 0.606) + (0.4 * 0.8333) + (0.2 * 0.8333) = 0.2424 + 0.3333 + 0.1666 = 0.724$$

**Una vez valorados los sub-atributos, se puede evaluar el atributo EFECTIVIDAD.**

$$EFECTIVIDAD = (C1 * EPP) + (C2 * EEP)$$

La organización ha definido las siguientes ponderaciones: C1 = 0.5 y C2 = 0.5.

$$EFECTIVIDAD = (0.5 * 0.3) + (0.5 * 0.724) = 0.15 + 0.362 = 0.512$$

**Evaluación atributo EFICIENCIA**

Para el atributo de EFICIENCIA se realiza el cálculo igual que para el atributo de efectividad los datos serían los siguientes:

$$EFICIENCIA = (C1 * ER) + (C2 * ET) + (C3 * EP)$$

La organización ha definido las siguientes ponderaciones:

$$C1 = 0.35, C2 = 0.30 \text{ y } C3 = 0.35$$

$$EFICIENCIA = (0.35 * 0.4) + (0.30 * 0.75) + (0.35 * 0.325) = 0.14 + 0.225 + 0.1138 = 0.478$$

**Tabla de valoración del atributo sugerida (se puede usar para todos):**

[Tomando como referencia la escala de calificación establecida en la norma ISO/IEC 15504]

*Tabla. Valorización sugerida para Efectividad*

Valor alcanzado por el atributo	Valoración cualitativa
x > 0.85	Plenamente alcanzado
0.5 < x <= 0.85	Ampliamente alcanzado
0.15 < x <= 0.5	Parcialmente alcanzado
X < 0.15	NO alcanzado

• **Tabla de evaluación de proceso diligenciada:**

Esta tabla es entrada de la actividad P3.A3. y nos sirve como elemento para elaborar reporte y divulgar resultados

		ATRIBUTOS DE PROCESO DE PRUEBA				
		0.512 EFECTIVIDAD		0.478 EFICIENCIA		
		0.5 Preparación de Pruebas EPP	0.5 Ejecución de Pruebas EEP	0.35 Recursos ER	0.30 Tiempo ET	0.35 Productividad EP
Categoría (y sus Métodos)	Estáticas					
	Unitarias	0.3	0.724	0.4	0.75	0.325
	Integración					
	Aceptación					
	Sistema					
Documentación	Plan de Prueba					
	Casos de Prueba					
	Reportes					

• **Reporte de evaluación de proceso de prueba**

Este reporte es la única salida de la actividad P3.A3. se puede realizar a partir de la tabla de evaluación de proceso diligenciada

La valoración del atributo EFECTIVIDAD fue de 0.512 lo que indica que este atributo fue Ampliamente alcanzado, sin embargo, la valoración del atributo EFICIENCIA fue de 0.478 lo que indica que el atributo fue Parcialmente alcanzado.

Para el atributo EFICIENCIA la organización ha determinado que no se han cumplido con los niveles de satisfacción requeridos por lo tanto se deberán seguir los protocolos establecidos para solucionar este problema.

Con la elaboración del reporte de evaluación de proceso de prueba se da por terminado el ejemplo de uso de la guía. Se insiste que todos estos procesos se pueden aplicar cuando se requiera evaluar actividades de testing para diversas categorías y en diferentes momentos del proyecto de desarrollo de software.

La aplicación frecuente de los procesos y actividades presentados en esta guía ofrecen una oportunidad para que una pequeña empresa de software pueda valorar el estado de sus actividades de testing.

## **ANEXO D**

### **Documentación para evaluar la guía técnica**

## ANEXO D.

### Documentación para evaluar la guía técnica

En este anexo se presentan los documentos y/formatos que se elaboraron para evaluar la propuesta de la guía técnica mediante el método sesión de grupo.

#### D.1. Protocolo para la sesión de grupo

Tema: Evaluación de la propuesta de una guía técnica para la medición y valoración en actividades de testing para pequeñas organizaciones desarrolladoras de software.

Duración de la sesión: 60 – 120 minutos

Método de la sesión: virtual

Fecha: mayo 11 de 2021

Hora: 9:00 a.m

Moderador(es): Javier Eduardo Pino Belalcazar ([javierbelalcaza@unicauca.edu.co](mailto:javierbelalcaza@unicauca.edu.co))

Jennifer Andrea Figueroa Bolaños ([andrefb@unicauca.edu.co](mailto:andrefb@unicauca.edu.co))

Supervisor: Carlos Alberto Ardila Albarracín ([cardila@unicauca.edu.co](mailto:cardila@unicauca.edu.co))

Materiales para la sesión:

1. Documento sintetizado sobre la guía técnica
2. Ficha de participante
3. Documento compacto sobre la guía técnica
4. Formato de evaluación de componentes
5. Encuesta de evaluación de aspectos generales

Orden de la sesión:

1. Diligenciamiento de ficha de participante
2. Presentación de la guía técnica
3. Discusión sobre la guía técnica
4. Diligenciamiento del formato de evaluación de componentes
5. Diligenciamiento de la encuesta de evaluación de aspectos generales

## D.2. Ficha de participante de la sesión de grupo

La información que se requiere de los participantes de la sesión de grupo se presenta en la Tabla 1.

<b>FICHA DE PARTICIPANTE EN SESION DE FOCUS GROUP</b>	
Tema	Evaluación de la propuesta de una guía técnica para la medición en actividades de testing para pequeñas organizaciones desarrolladoras de software
Nombre completo	
Formación académica	
Organización	
Actividad principal de la organización	
Cargo	
Información de contacto (correo electrónico y/o teléfono)	
Especifique experiencia en actividades de testing	

Table 1. Ficha de participante

### D.3. Ficha de participante de la sesión de grupo

Por favor indique en el formato que se presenta a continuación (Tabla 1) aspectos positivos, aspectos a mejorar y/o observaciones que usted considere deban incluirse en relación con los componentes de la guía técnica

<b>Componentes de la Guía Técnica</b>	<b>Aspectos Positivos</b>	<b>Aspectos a mejorar</b>	<b>Observaciones</b>
Repositorio de Métricas			
Tabla de evaluación de procesos			
Ejemplo de uso para la guía			

Table 2. Formato de evaluación de componentes

## D.4. Encuesta de evaluación de aspectos generales sobre guía técnica

1. ¿Considera usted que la guía técnica propuesta es necesaria para las pequeñas organizaciones desarrolladoras de software?

SÍ \_\_\_ NO \_\_\_

2. ¿Considera usted que los componentes de la guía técnica son adecuados para las pequeñas organizaciones desarrolladoras de software?

SÍ \_\_\_ NO \_\_\_

3. ¿Considera usted que las actividades propuestas en la guía técnica son apropiadas para pequeñas organizaciones desarrolladoras de software?

SÍ \_\_\_ NO \_\_\_

4. ¿Considera usted que los productos generados en las actividades son adecuados para pequeñas organizaciones desarrolladoras de software?

SÍ \_\_\_ NO \_\_\_

5. ¿Considera usted que las salidas están acordes con cada actividad indicada en la guía técnica?

SÍ \_\_\_ NO \_\_\_

6. ¿Considera usted que las descripciones presentadas muestran de forma clara la estructura de la guía técnica y las actividades propuestas?

SÍ \_\_\_ NO \_\_\_

7. ¿Considera usted que el repositorio de métricas de la guía técnica es claro y conciso?

SÍ \_\_\_ NO \_\_\_

8. ¿Considera usted que la tabla de evaluación de procesos de la guía técnica es clara y concisa?

SÍ \_\_\_ NO \_\_\_

9. ¿Considera usted que la relación entre las métricas y los sub-atributos de eficiencia y eficacia son claros y concisos?

SÍ \_\_\_ NO \_\_\_

10. ¿Considera usted que le hacen falta elementos a la guía técnica? En caso afirmativo, por favor mencionarlos.

SÍ \_\_\_ NO \_\_\_

11. ¿Considera usted que hay debilidades en la guía técnica? En caso afirmativo, por favor mencionarlas.

SÍ \_\_\_ NO \_\_\_

12. ¿De acuerdo a su experiencia, ¿considera usted que la guía técnica puede ser de fácil aplicación?

SÍ \_\_\_ NO \_\_\_