

PROCEDIMIENTO DE ANÁLISIS CAUSAL ENFOCADO EN PEQUEÑAS ORGANIZACIONES DE DESARROLLO SOFTWARE



ANEXOS

Kelly Yohanna Zuñiga Silva

Monografía para optar al título de

Ingeniero de Sistemas

Director: PhD. Francisco José Pino Correa

Co-director: Mag. Carlos Alberto Ardila Albarracín

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Departamento de Sistemas – Grupo de Investigación IDIS

Línea de Investigación calidad de proceso y producto

Popayán, Marzo de 2015

TABLA DE ANEXOS

ANEXO 1. PLANTILLAS	6
ANEXO 2. COMPARACIÓN DE TÉCNICAS EMPLEADAS EN EL ANÁLISIS CAUSAL ...	25
ANEXO 3. ESCALAS PARA EL ANÁLISIS MODAL DE FALLA Y EFECTO.....	57
ANEXO 4. TRADUCCIÓN GUÍA IBM.....	59
ANEXO 5. ENTREVISTAS A LOS INVOLUCRADOS EN EL PROYECTO.....	69
ANEXO 6. ENTREVISTAS PARA EL CLIENTE.....	72
ANEXO 7. RECOLECCIÓN DE DATOS	77

LISTA DE TABLAS

Tabla 1. Análisis - Diagrama de Ishikawa	26
Tabla 2. Análisis Diagrama de Pareto.....	28
Tabla 3. Análisis mapas mentales	30
Tabla 4. Análisis pensamiento sistémico.....	32
Tabla 5. Análisis mapas causales.....	34
Tabla 6. Análisis mapas conceptuales.....	36
Tabla 7. Análisis Modelo de ecuaciones estructurales.....	38
Tabla 8. Análisis modelos gráficos	39
Tabla 9. Análisis de trayectoria o de ruta	41
Tabla 10. Análisis de gráfico radial.....	43
Tabla 11. Modelo basado en reglas	45
Tabla 12. Análisis – teoría relacional para mapas causales	46
Tabla 13. Análisis – Diagrama de dispersión o de correlación	47
Tabla 14. Diagrama de afinidad.....	49
Tabla 15. Análisis modal de fallo y efectos	50
Tabla 16. Criterio- Referencias encontradas de su aplicación dentro de la ingeniería del software	52
Tabla 17. Criterio-Nivel de conocimientos requeridos para el uso de la técnica.....	52
Tabla 18. Criterio-Nivel de complejidad en la implementación del procedimiento.....	52
Tabla 19. Criterio-Nivel de complejidad en la interpretación del esquema	53
Tabla 20. Criterio-Herramientas de software libre disponibles	53
Tabla 21. Criterio-Nivel de conocimientos estadísticos.....	53

Tabla 22. Calificación de las herramientas para análisis causal	55
Tabla 23. Ponderación de los criterios para evaluar las herramientas de análisis causal	56
Tabla 24. Escala tradicional de ocurrencia (O)	57
Tabla 25. Escala de clasificación de gravedad (S)	58
Tabla 26. Escala tradicional para la detección (D).....	58
Tabla 27. Etapa de apertura ODC	64
Tabla 28. Etapa de cierre ODC	68
Tabla 29. Plantilla de recolección de defectos, fase 1	79
Tabla 30. Plantilla de recolección de defectos, fase 2	97

LISTA DE FIGURAS

Figura 1. Diagrama de Pareto para los defectos detectados en la fase 1	83
Figura 2. Diagrama de Ishikawa D1.4.....	85
Figura 3. Diagrama de Ishikawa D1.1	85
Figura 4. Diagrama de Ishikawa D1.2.....	86
Figura 5. Diagrama de Ishikawa D1.3.....	86
Figura 6. Diagrama de Ishikawa D1.5.....	87
Figura 7. Diagrama de Ishikawa D1.8.....	87
Figura 8. Diagrama de Ishikawa D1.7	88
Figura 9. Diagrama radial fase 1	90
Figura 10. Diagrama de Pareto para los defectos detectados en la fase 2	98
Figura 11. Diagrama de Ishikawa D2.7.....	105
Figura 12. Diagrama de Ishikawa D2.2.....	105
Figura 13. Diagrama de Ishikawa D2.6.....	106
Figura 14. Defectos relacionados con el producto software.....	107
Figura 15. Diagrama radial fase 2.....	109

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

ANEXO 1. PLANTILLAS

1. Actividad Preparación

1.1. Plantilla del perfil del aspirante al grupo de análisis causal

PLANTILLA DEL PERFIL DEL ASPIRANTE AL GRUPO DE ANÁLISIS CAUSAL	
PROCEDIMIENTO DE ANÁLISIS CAUSAL ENFOCADO EN PEQUEÑAS ORGANIZACIONES DE DESARROLLO SOFTWARE (PAC-DS)	
Organización	<i>[nombre de la organización]</i>
Proyecto	<i>[nombre del Proyecto]</i>
Nombre: <i>[En esta sección se escribe el nombre del candidato al grupo de análisis causal]</i>	
Competencias	
<i>[En esta sección el candidato lista sus competencias]</i>	
Experiencia laboral	
<i>[En esta sección el candidato escribe experiencias donde ha aplicado/desarrollado sus competencias: lugar, hace cuanto, duración, observaciones.]</i>	
Elaboró: _____ Fecha: _____	Revisó: _____ Fecha: _____

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

1.2. Plantilla del perfil del líder de análisis causal

PLANTILLA DEL PERFIL DEL LÍDER DE ANÁLISIS CAUSAL			
PROCEDIMIENTO DE ANÁLISIS CAUSAL ENFOCADO EN PEQUEÑAS ORGANIZACIONES DE DESARROLLO SOFTWARE (PAC-DS)			
Organización	<i>[nombre de la organización]</i>		
Proyecto	<i>[nombre del Proyecto]</i>		
Nombre: <i>[En esta sección se escribe el nombre del candidato a ser líder de análisis causal]</i>			
Competencias			
<i>[En esta sección el candidato lista sus competencias]</i>			
Experiencia laboral			
<i>[En esta sección el candidato escribe experiencias donde ha aplicado/desarrollado sus competencias: lugar, hace cuanto, duración, observaciones.]</i>			
Conocimientos para el análisis causal		Si	No
<i>El candidato tiene conocimiento de técnica empleadas para el análisis causal, por ejemplo diagrama de Ishikawa, diagrama de Pareto, mapas causales, entre otros.</i>			
<i>El candidato conoce el funcionamiento de la clasificación ODC (Orthogonal Defect Classification)</i>			
<i>El candidato tiene conocimientos sobre la gestión de equipos</i>			
<i>[la persona encargada de elegir al líder de análisis causal puede agregar otros conocimientos que considere importantes]</i>			
Elaboró: _____ Fecha: _____	Revisó: _____ Fecha: _____		

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

2. Actividad Detección de defectos

2.1. Plantilla de recolección de defectos

PLANTILLA DE RECOLECCIÓN DE DEFECTOS		
PROCEDIMIENTO DE ANÁLISIS CAUSAL ENFOCADO EN PEQUEÑAS ORGANIZACIONES DE DESARROLLO SOFTWARE (PAC-DS)		
Organización	<i>[nombre de la organización]</i>	
Proyecto	<i>[nombre del Proyecto]</i>	
Fecha: <i>[En esta sección se escribe la fecha de la reunión de análisis causal en que se detectó el(los) defecto(s)]</i>		
ID	Defecto	Cantidad de personas que lo reportaron
<i>[En esta sección se escribe el identificador del defecto]</i>	<i>[En esta sección se escribe el defecto encontrado]</i>	<i>[En esta sección se escribe la cantidad de personas que reportaron el defecto durante la reunión de análisis causal]</i>
DIAGRAMA DE AFINIDAD		
<i>[En esta sección se coloca el diagrama de afinidad resultante de los defectos analizados]</i>		
Elaboró: _____ Fecha: _____	Revisó: _____ Fecha: _____	

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

2.2. Plantilla de defectos de mayor impacto

Ficha del documento

Fecha	Revisión	Autor	Observaciones
<i>[Se especifica la fecha en que se emite el documento]</i>	<i>[Un registro histórico de las acciones realizadas sobre el documento, lo cual permite gestionar los cambios y en un momento específico tomar medidas pertinentes]</i>	<i>[Se especifica el nombre de la persona encargada de realizar el documento].</i>	<i>[Se especifica consideraciones importantes a tener en cuenta]</i>
Elaboró: _____ Fecha: _____		Revisó: _____ Fecha: _____	

Propósito

El propósito de este documento es mantener evidencias de los defectos que generan mayor impacto en el desarrollo del proyecto.

El documento contiene:

1. Diagrama de Pareto o análisis modal de fallo y efecto

[En esta sección se debe colocar el diagrama de Pareto obtenido a partir de los defectos recolectados en la reunión de análisis causal o/y se realiza el análisis modal de fallo y efecto]

2. Defectos de mayor impacto

[Esta sección contiene los defectos considerados de mayor impacto para el desarrollo del proyecto. Éstos son obtenidos teniendo como referente el Diagrama de Pareto y/o análisis modal de fallo y defecto.]

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

Identificador	Defecto
<i>[Los defectos son listados de mayor a menor, según los resultados obtenidos en el diagrama de Pareto y/o análisis modal de fallo y efecto].</i>	<i>[Se describe el defecto]</i>

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

2.3. Plantilla para clasificar los defectos

Ficha del documento

Fecha	Revisión	Autor	Observaciones
<i>[Se especifica la fecha en que se emite el documento]</i>	<i>[Un registro histórico de las acciones realizadas sobre el documento, lo cual permite gestionar los cambios y en un momento específico tomar medidas pertinentes]</i>	<i>[Se especifica el nombre de la persona encargada de realizar el documento].</i>	<i>[Se especifica consideraciones importantes a tener en cuenta]</i>
Elaboró: _____ Fecha: _____		Revisó: _____ Fecha: _____	

Propósito

El propósito de este documento es mantener evidencias de los defectos presentados y clasificados de acuerdo al esquema de clasificación ODC (Orthogonal Defect Classification).

El documento contiene:

1. Clasificación de los defectos

[Cada defecto debe ser clasificado de acuerdo a la tabla mostrada a continuación]

Defecto: <i>[En esta sección se escribe el defecto encontrado]</i>			
Fecha:		Fecha:	
Etapas de apertura		Etapas de cierre	
Actividad	<i>[Actividad que se estaba realizando al</i>	Origen	<i>[se refiere a que entidad debe ser corregida (la</i>

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

	<i>encontrar el defecto]</i>		<i>ubicación del defecto)]</i>
Disparador	<i>[Se refiere a como se detecta el defecto]</i>	Tipo de defecto	<i>[Se refiere a que debe ser corregido]</i>
Impacto para el cliente	<i>[se refiere al impacto que tiene el defecto sobre el cliente]</i>	Calificador	<i>[Explica cómo debe ser corregido el defecto que se encontró]</i>
		Fuente	<i>[Captura el origen del código que tenía el defecto (Target)]</i>
		Edad	<i>[Se refiere al historial del origen]</i>

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

2.4. Plantilla del documento de defectos

Ficha del documento

Fecha	Revisión	Autor	Observaciones
<i>[Se especifica la fecha en que se emite el documento]</i>	<i>[Un registro histórico de las acciones realizadas sobre el documento, lo cual permite gestionar los cambios y en un momento específico tomar medidas pertinentes]</i>	<i>[Se especifica el nombre de la persona encargada de realizar el documento].</i>	<i>[Se especifica consideraciones importantes a tener en cuenta]</i>
Elaboró: _____ Fecha: _____		Revisó: _____ Fecha: _____	

Propósito

El propósito de este documento es condensar la información recolectada en la actividad Detección de defectos

El documento contiene:

1. Defectos identificados

[En esta sección se debe colocar la información obtenida en la tarea Identificar los defectos]

2. Muestra de los defectos

[En esta sección se debe colocar la información obtenida en la tarea Determinar muestra de los defectos]

3. Clasificación de los defectos

[En esta sección se debe colocar la información obtenida en la tarea Clasificar los defectos]

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

3. Actividad detección de causas fundamentales

3.1. Plantilla de causas encontradas

Ficha del documento

Fecha	Revisión	Autor	Observaciones
<i>[Se especifica la fecha en que se emite el documento]</i>	<i>[Un registro histórico de las acciones realizadas sobre el documento, lo cual permite gestionar los cambios y en un momento específico tomar medidas pertinentes]</i>	<i>[Se especifica el nombre de la persona encargada de realizar el documento].</i>	<i>[Se especifica consideraciones importantes a tener en cuenta]</i>
Elaboró: _____ Fecha: _____		Revisó: _____ Fecha: _____	

Propósito

El propósito de este documento es tener información histórica que permita comparar situaciones perjudiciales presentadas previamente y evitar su aparición en proyectos futuros. Asimismo este reporte especifica el responsable de realizar los diagramas correspondientes.

El documento contiene:

1. Diagramas

[En esta sección el autor del documento debe colocar todos los diagramas realizados durante la reunión con el grupo de análisis causal]

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

2. Síntesis de las causas clasificadas en categorías

	Categorías				Otra categoría
Defecto	Herramientas	Entradas	Personas	Métodos	

[Se debe seleccionar con un X la categoría a la que pertenece el defecto]

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

3.2. Plantilla de causas de fundamentales

Ficha del documento

Fecha	Revisión	Autor	Observaciones
<i>[Se especifica la fecha en que se emite el documento]</i>	<i>[Un registro histórico de las acciones realizadas sobre el documento, lo cual permite gestionar los cambios y en un momento específico tomar medidas pertinentes]</i>	<i>[Se especifica el nombre de la persona encargada de realizar el documento].</i>	<i>[Se especifica consideraciones importantes a tener en cuenta]</i>
Elaboró: _____ Fecha: _____		Revisó: _____ Fecha: _____	

Propósito

El propósito de este documento es mantener evidencias de las causas que generan mayor impacto en el desarrollo del proyecto.

El documento contiene:

1. Causas fundamentales

[En esta sección se debe colocar las causas que deben ser eliminadas, para ello se debe considerar el defecto tratado y su causa]

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

Defecto	Causa

2. Gráfico radial

[En esta sección se debe colocar en gráfico radial]

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

3.3. Plantilla de recomendaciones

PLANTILLA DE RECOMENDACIONES				
PROCEDIMIENTO DE ANÁLISIS CAUSAL ENFOCADO EN PEQUEÑAS ORGANIZACIONES DE DESARROLLO SOFTWARE (PAC-DS)				
Organización	<i>[nombre de la organización]</i>			
Proyecto	<i>[nombre del Proyecto]</i>			
Autor de las recomendaciones: <i>[En esta sección se escribe el nombre de la persona encargada de realizar las recomendaciones.]</i>				
Beneficios esperados al implementar las recomendaciones				
<i>[En esta sección se lista los beneficios esperados al implementar las recomendaciones]</i>				
ID	Causa	Recomendaciones	Responsable	Fecha límite
<i>[ID de la recomendación]</i>	<i>[En esta sección se escribe una breve descripción de la causa.]</i>	<i>[En esta sección se escribe la o las recomendaciones dadas para eliminar la causa.]</i>	<i>[En esta sección se escribe el nombre de la persona encargada de implementar los cambios propuestos.]</i>	<i>[En esta sección se escribe la fecha límite que tiene el responsable para implementar el cambio.]</i>
Elaboró: _____		Revisó: _____		
Fecha: _____		Fecha: _____		

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

3.4. Plantilla de solicitud de cambio

PLANTILLA DE SOLICITUD DE CAMBIOS			
PROCEDIMIENTO DE ANÁLISIS CAUSAL ENFOCADO EN PEQUEÑAS ORGANIZACIONES DE DESARROLLO SOFTWARE (PAC-DS)			
Organización	<i>[nombre de la organización]</i>		
Proyecto	<i>[nombre del Proyecto]</i>		
Fecha de solicitud		Fecha de recepción	
Justificación del cambio propuesto			
<i>[En esta sección se escriben los motivos por los cuales es necesario realizar el cambio.]</i>			
Artefactos afectados por el cambio			
Nombre del artefacto	Observaciones		
<i>[En esta sección se listan los artefactos afectados por el cambio.]</i>	<i>[En esta sección se explica cómo se ve afectado el artefacto por el cambio solicitado.]</i>		
Consecuencias de no realizar el cambio			
<i>[En esta sección se escriben los aspectos negativos que han surgido o surgirán como consecuencia de no realizar el cambio solicitado.]</i>			
Cambio	Aprobado	<input type="checkbox"/>	
	Rechazado	<input type="checkbox"/>	
	Pospuesto	<input type="checkbox"/>	
Responsable de aplicar el cambio			
Elaboró: _____	Revisó: _____		
Fecha: _____	Fecha: _____		

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

3.5. Plantilla de seguimiento de recomendaciones

PLANTILLA DE SEGUIMIENTO DE RECOMENDACIONES			
PROCEDIMIENTO DE ANÁLISIS CAUSAL ENFOCADO EN PEQUEÑAS ORGANIZACIONES DE DESARROLLO SOFTWARE (PAC-DS)			
Organización	<i>[nombre de la organización]</i>		
Proyecto	<i>[nombre del Proyecto]</i>		
ID	Fecha	Verificación del cumplimiento	Fecha de cierre
<i>[Identificador de la recomendación.]</i>	<i>[Fecha en que se verificó el cumplimiento de la recomendación]</i>	<i>[En esta sección se describe los cambios realizados]</i>	<i>[En esta sección se escribe la fecha en que se ejecutó a totalidad la recomendación]</i>
Elaboró: _____ Fecha: _____		Revisó: _____ Fecha: _____	

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

3.6. Plantilla del documento de causas

Ficha del documento

Fecha	Revisión	Autor	Observaciones
<i>[Se especifica la fecha en que se emite el documento]</i>	<i>[Un registro histórico de las acciones realizadas sobre el documento, lo cual permite gestionar los cambios y en un momento específico tomar medidas pertinentes]</i>	<i>[Se especifica el nombre de la persona encargada de realizar el documento].</i>	<i>[Se especifica consideraciones importantes a tener en cuenta]</i>
Elaboró: _____ Fecha: _____		Revisó: _____ Fecha: _____	

Propósito

El propósito de este documento es condensar la información recolectada en la actividad Detección de causas fundamentales

El documento contiene:

1. Causas identificadas

[En esta sección se debe colocar la información obtenida en la tarea Identificar las causas]

2. Análisis de causas

[En esta sección se debe colocar la información obtenida en la tarea Analizar las causas]

3. Recomendaciones

[En esta sección se debe colocar la información obtenida en la tarea Desarrollar recomendaciones]

4. Solicitudes de cambios

[En esta sección se debe colocar la información obtenida en la tarea Iniciar solicitud de cambios]

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

5. Seguimiento a las recomendaciones

[En esta sección se debe colocar la información obtenida en la tarea Hacer seguimiento a las recomendaciones]

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

4. Actividad Documentar

4.1. Plantilla de lecciones aprendidas

PLANTILLAS DE LECCIONES APRENDIDAS					
PROCEDIMIENTO DE ANÁLISIS CAUSAL ENFOCADO EN PEQUEÑAS ORGANIZACIONES DE DESARROLLO SOFTWARE (PAC-DS)					
Organización	<i>[nombre de la organización]</i>				
Proyecto	<i>[nombre del Proyecto]</i>				
Fecha de Inicio	<i>[Fecha de inicio el proyecto]</i>		Fecha fin	<i>[Fecha fin del proyecto]</i>	
Líder del proyecto	<i>[Nombre del responsable del proyecto]</i>				
Cliente final	<i>[Nombre del cliente para el cual se desarrolló el proyecto]</i>				
Miembros del equipo			Rol		
<i>[Nombre de todos los miembros del proyecto]</i>					
ID	Lección aprendida	Fase del proyecto	Recomendaciones implementadas	Resultados obtenidos	Observaciones
<i>[identificador único de la lección]</i>	<i>[Se describe la lección aprendida]</i>	<i>[Disciplina del ciclo de vida del software donde se presentó la]</i>	<i>[Se describe las decisiones tomadas y acciones implementadas para]</i>	<i>[Se describe los resultados positivos y negativos obtenidos al implementar las]</i>	<i>[se describe las recomendaciones que se deben implementar en proyectos similares, así]</i>

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

<i>aprendida]</i>		<i>lección]</i>	<i>la situación presentada]</i>	<i>recomendaciones]</i>	<i>como aquellas situaciones que se deben evitar y otras alternativas que se pueden aplicar]</i>
Elaboró: _____ Fecha: _____			Revisó: _____ Fecha: _____		

Esta plantilla ha sido adaptada de la plantilla mostrada en: <http://goo.gl/5cbQ95>

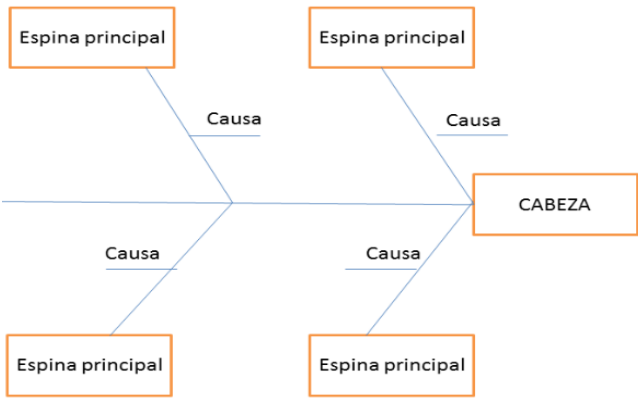
Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

ANEXO 2. COMPARACIÓN DE TÉCNICAS EMPLEADAS EN EL ANÁLISIS CAUSAL

1. Técnicas para análisis causal

A continuación se realiza una investigación sobre las técnicas que pueden ser útiles para apoyar el análisis causal.

1.1. Diagrama de Ishikawa

DIAGRAMA DE ISHIKAWA
ESQUEMA  <p>El diagrama muestra una estructura de causa-efecto. A la izquierda, una línea horizontal se divide en cuatro ramas diagonales que apuntan hacia la derecha. Cada rama termina en un recuadro etiquetado como 'Espina principal'. A la derecha de estas ramas, una línea horizontal converge y termina en un recuadro etiquetado como 'CABEZA'. Las líneas que conectan las 'Espina principal' con la 'CABEZA' están etiquetadas como 'Causa'.</p>
HERRAMIENTAS SOFTWARE En internet se pueden encontrar programas que permiten realizar diagramas de Ishikawa, a continuación se menciona algunos de ellos. <ul style="list-style-type: none">❖ Cmap Tools❖ OpenOffice.org❖ Microsoft PowerPoint
DESCRIPCIÓN El diagrama de Ishikawa (también conocido como diagrama de espina de pescado o diagrama Causa-Efecto) fue inventado por Kaoru Ishikawa, quien fue un teórico de la administración de empresas japonés, experto en el control de calidad. Este diagrama muestra las causas de un determinado evento y es considerado como una de las siete herramientas básicas del control de calidad. El diagrama tiene forma de espina de pescado, donde el problema principal es representado en la cabeza y las causas potenciales, que pueden ser obtenidas a partir de investigación o haciendo uso de técnicas que permiten encontrar la causa del problema, se indican en los huesos del pescado [1]. Asimismo el diagrama permite visualizar relaciones entre elementos que aparentemente no tenían relación alguna [2]. Esta herramienta es utilizada para mejorar la calidad en los procesos [3].
VENTAJAS <ul style="list-style-type: none">❖ Su implementación requiere un mínimo de capacitación, previa al uso de la herramienta [4].❖ Ayuda a mejorar la calidad y la productividad en el desarrollo del software [5].

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

<ul style="list-style-type: none"> ❖ Proporciona información en cualquier etapa del proceso de desarrollo [5]. ❖ Se muestra a los desarrolladores el valor de la conformidad del proceso [5].
DESVENTAJAS
<ul style="list-style-type: none"> ❖ No se tiene definido un punto de inicio y fin en la cadena de causa y efecto [5]. ❖ El diagrama de Ishikawa no gestiona situaciones en las cuales puede haber una conexión en cascada y darse bucles complejos [5]. ❖ Los defectos del software deben estar debidamente clasificados, esto para evitar fallos humanos, que podrían generarse a partir de interpretaciones erróneas sobre la causa real [5]. ❖ Al dar prioridad a los defectos individuales, se reduce la atención a la búsqueda de soluciones frente a un ámbito más amplio de problemas [5]. ❖ Se debe tener cuidado en el por qué de las causas, verificando las acciones y condiciones que influyen en la causa, de lo contrario se hará saltos en la lógica [4].
PROCEDIMIENTO
<p>Su representación consiste en un rectángulo situado a la derecha del esquema donde se indica el efecto que se quiere analizar, se dibuja una fecha de entrada (a modo de columna vertebral de pescado) a este rectángulo, a donde llegaran las otras flechas provenientes de los posibles focos de los problemas que generan el efecto que se esta estudiando. A estas fechas le llegaran otras secundarias con posibles sub-causas relacionadas con dichos focos [6].</p> <p>A continuación se describe el procedimiento planteado en el libro calidad de sistemas informáticos de los autores Mario Piattini, Félix García e Ismael Caballero [6]:</p> <ol style="list-style-type: none"> 1. Elaborar un enunciado claro del efecto. 2. Dibujar el diagrama de la espina de pescado, colocando el efecto (problema) en el cuadro al lado derecho. 3. Identificar de 3 a 6 espinas mayores que puedan ser las causas del problema/efecto principal. 4. Dibujar las espinas mayores como fechas inclinadas dirigidas a la fecha principal. 5. Identificar causas de primer nivel relacionadas con cada espina mayor. 6. Identificar causas de segundo nivel para cada causa de primer nivel. 7. Identificar causas de tercer nivel, para cada causa de segundo nivel, y así sucesivamente. 8. Observando los resultados, identificar la causa raíz que permita obtener conclusiones en la resolución del problema.

Tabla 1. Análisis - Diagrama de Ishikawa

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

1.2. Diagrama de Pareto

DIAGRAMA DE PARETO																	
ESQUEMA																	
<table border="1"> <caption>Datos del Diagrama de Pareto</caption> <thead> <tr> <th>Problema</th> <th>Frecuencia</th> </tr> </thead> <tbody> <tr> <td>Problema1</td> <td>80</td> </tr> <tr> <td>Problema2</td> <td>70</td> </tr> <tr> <td>Problema3</td> <td>60</td> </tr> <tr> <td>Problema4</td> <td>50</td> </tr> <tr> <td>Problema5</td> <td>40</td> </tr> <tr> <td>Problema6</td> <td>30</td> </tr> <tr> <td>Problema7</td> <td>20</td> </tr> </tbody> </table>		Problema	Frecuencia	Problema1	80	Problema2	70	Problema3	60	Problema4	50	Problema5	40	Problema6	30	Problema7	20
Problema	Frecuencia																
Problema1	80																
Problema2	70																
Problema3	60																
Problema4	50																
Problema5	40																
Problema6	30																
Problema7	20																
HERRAMIENTAS SOFTWARE																	
<p>En internet se pueden encontrar programas que permiten realizar diagramas de Pareto, a continuación se menciona algunos de ellos.</p> <ul style="list-style-type: none"> ❖ Microsof Excel ❖ Pareto Pro ❖ QI Macros ❖ Apache OpenOffice Calc ❖ LibreOffice Calc 																	
DESCRIPCIÓN																	
<p>El diagrama de Pareto es considerado como una de las herramientas más importantes en la mejora de la calidad del proceso. Este diagrama es una representación gráfica de los problemas del proceso y da una idea del orden en que deben abordarse estos problemas, pues la clasificación se hace de acuerdo a los problemas que se presentan con mayor incidencia, hasta aquellos que se presentan con menor frecuencia, en orden descendente de izquierda a derecha [7]. Establece que “al eliminar el 20% de las causas que generan un problema en una situación resolvería el 80% de ellos, mientras que el 80% de las causas restantes resuelven el 20% de los problemas restantes”.</p> <p>El diagrama de Pareto es una herramienta importante en la mejora de los procesos de calidad [7] [8].</p>																	
VENTAJAS																	
<ul style="list-style-type: none"> ❖ El diagrama de Pareto permite concentrarse en las causas que generan mayor impacto [7] [8]. ❖ Mediante el gráfico se puede interpretar de forma más rápida la importancia de los problemas [7]. ❖ Es posible comparara diagramas obtenidos a lo largo de un proyecto, con el fin de comparar las mejoras realizadas [7]. 																	
DESVENTAJAS																	
<ul style="list-style-type: none"> ❖ Es difícil identificar el 20% de las causas que están ocasionando la mayoría de los problemas [9]. 																	

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

<ul style="list-style-type: none">❖ En ocasiones el personal solo se centra en los problemas principales, sin tener en cuenta el impacto futuro, de los problemas triviales [9].❖ No da una idea clara sobre la causa raíz que está generando los problemas.
PROCEDIMIENTO
<p>A continuación se describe el procedimiento planteado en el libro calidad de sistemas informáticos de los autores Mario Piattini, Félix García e Ismael Caballero [6]:</p> <ol style="list-style-type: none">1. Identificar el problema a analizar, seleccionando los problemas o variables que se van a investigar, diciendo los datos y la forma de clasificarlos y definiendo el método a utilizar en la recopilación de datos.2. Diseñar una hoja de recopilación de datos para que guarden datos sobre las causas a investigar y el número de veces que aparecen.3. Reunir los datos y efectuar el cálculo de porcentaje de las frecuencias de aparición.4. Ordenar los datos en orden decreciente de frecuencia.5. Una vez en esta disposición, calcular las frecuencias acumuladas para cada causa.6. Dibujar dos ejes verticales y un eje horizontal. Marcar el eje vertical izquierdo con la cantidad de causas acumuladas y el derecho con una escala del 0% hasta 100%. Luego se divide el eje horizontal en un número de intervalos igual al número de ítems clasificados.7. Construir un gráfico de barras con el mismo ancho y sin dejar espacio entre ellas, este gráfico está basado en las cantidades y porcentajes de cada uno de los ítems colocándolos de mayor a menor y de izquierda a derecha.8. Dibujar la curva de frecuencias acumuladas.9. Escribir cualquier información necesaria sobre el diagrama y sobre los datos.

Tabla 2. Análisis Diagrama de Pareto

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

1.3. Mapas mentales

MAPAS MENTALES
ESQUEMA
<pre> graph TD IP(Idea principal) --- I1(Idea 1) IP --- I2(Idea 2) IP --- I3(Idea 3) IP --- I4(Idea 4) I1 --- SI1.1(Sub-Idea 1.1) I1 --- SI1.2(Sub-Idea 1.2) I2 --- SI2.1(Sub-Idea 2.1) I2 --- SI2.2(Sub-Idea 2.2) I3 --- SI3.1(Sub-Idea 3.1) I3 --- SI3.2(Sub-Idea 3.2) I4 --- SI4.1(sub-Idea 4.1) I4 --- SI4.2(Sub-Idea 4.2) </pre>
HERRAMIENTAS SOFTWARE
<p>Existen aplicaciones en línea que permiten realizar mapas mentales, tales como examTime y MinMap, FreeMind.</p> <p>En [10] emplean Mind Manager para realizar mapas mentales.</p>
DESCRIPCIÓN
<p>“Un mapa mental es un diagrama usado para representar palabras, ideas, tareas u otros elementos vinculados y dispuestos radialmente alrededor de una palabra clave o idea central. Son empleados para generar, visualizar, estructurar y clasificar ideas, asimismo son de gran utilidad en el estudio, organización, solución de problemas y en la toma de decisiones” [11].</p> <p>En [11] se puede ver como los mapas mentales son aplicados como un enfoque que permite representar el análisis de causa raíz.</p>
VENTAJAS
<p>Según [12] los mapas mentales permiten la representación de ideas generadas a partir de un tema central, asimismo en la creación de los mapas mentales existen muy pocas reglas, la principal es contar con el cerebro y la imaginación, dando lugar a una organización que refleja la manera en que el cerebro organiza ideas. Por estos motivos realizar un mapa mental es un proceso sencillo que permite visualizar problemas complejos de las organizaciones. Además la persona encargada de la creación del mapa mental es libre de emplear su imaginación y crear el mapa sin necesidad de seguir un proceso sistemático. Asimismo es posible crear asociaciones creativas entre las ideas, lo cual facilita la detección de la relación existente entre cada una de ellas [13].</p> <p>Por otra parte el uso de imágenes y colores ayuda a las personas a entender y recordar conceptos con mayor facilidad en comparación a realizar estas actividades mediante otras técnicas, ya que es más fácil recordar un diagrama que una descripción [13].</p> <p>A continuación se menciona las ventajas mostradas en [10]:</p> <ul style="list-style-type: none"> ❖ Fácil para aprender y aplicar ❖ Fomenta la creatividad y la libre expresión ❖ Proporciona una visión general concisa jerárquica ❖ Fácil de ampliar y añadir más contenido

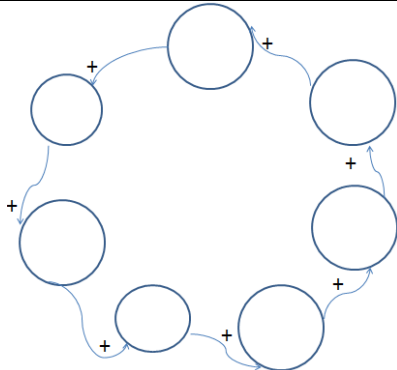
Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

DESVENTAJAS
<p>Las descripciones dadas en los mapas mentales pueden ser insuficientes, lo cual puede provocar que el lector no entienda de forma clara la idea plasmada [12]. Es por ello que el mapa mental es una guía visual en donde se representa una síntesis del tema, en lugar de representar todos los detalles referentes a la temática tratada [14].</p> <p>A continuación se menciona las desventajas mostradas en [10]</p> <ul style="list-style-type: none">❖ difícil de leer para aquellas personas que no estuvieron involucradas en el desarrollo del mapa mental❖ se enfoca en las relaciones jerárquicas❖ puede ser inconsistente❖ Puede llegar a ser excesivamente complejo
PROCEDIMIENTO
<p>Tomando como referencia el trabajo realizado en [15], se plantea el siguiente procedimiento:</p> <ol style="list-style-type: none">1. Comience en el centro con una imagen del tema, utilizando al menos 3 colores2. Dibuje en el centro de la hoja una imagen que simbolice la idea principal3. Partiendo de la imagen central irradia hacia el exterior las palabras clave las ideas más importantes relacionadas con el tema escogido4. Traza líneas curvas en lugar de rectas porque las líneas rectas aburren a tu cerebro.5. Las líneas deben estar conectados, a partir de la imagen central. Las líneas centrales son más gruesas, pero las líneas que irradian hacia fuera desde el centro deben ser más delgadas.6. Hacer las líneas de la misma longitud que la palabra / imagen que soportan.7. Utilice varios colores a lo largo del mapa mental, para la estimulación visual y también para codificar o agrupar.8. Desarrolle su propio estilo personal del mapa mental.9. Utilice asociaciones de énfasis para mostrar en su mapa mental.10. Mantener el mapa mental claro, usando jerarquía radial o contornos para abrazar las ramas.

Tabla 3. Análisis mapas mentales

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

1.4. Pensamiento sistémico/ Dinámica de sistemas

PENSAMIENTO SISTÉMICO/DINÁMICA DE SISTEMAS	
ESQUEMA	
	
HERRAMIENTAS SOFTWARE	
<ul style="list-style-type: none">❖ Vensim PLE (un software de simulación) permite modelar el pensamiento sistémico, es de bajo costo para uso comercial y gratuito para uso educativo.❖ Stella and iThink❖ Powersim❖ Software Evolucion	
DESCRIPCIÓN	
<p>Esta técnica fue desarrollada por Jay Forrester a mediados de 1950. Permite realizar una descripción cualitativa de sistemas complejos y es posible hacer simulación cuantitativa del comportamiento de éste [16].</p> <p>Los modelos de dinámica de sistemas ayudan a descubrir y representar los procesos de retroalimentación, así como los procesos complejos que determinan la dinámica del sistema, esto mediante el uso de ciclos [17]. Esta herramienta contribuye en el conocimiento de las características y la interrelación entre las variables que hacen parte del sistema. Además proporciona una base para las decisiones relativas al control, gestión, adquisición o transformación del sistema investigado. En este sentido la técnica es empleada para modelar sistemas complejos no lineales [18].</p> <p>Según [19] la dinámica de sistemas ha sido aplicada dentro de la ingeniería del software para mostrar el funcionamiento de los procesos.</p>	
VENTAJAS	
<p>De los artículos revisados se pueden destacar las siguientes ventajas:</p> <ul style="list-style-type: none">❖ El pensamiento sistémico permite entender el sistema de forma integral, lo cual ayuda a encontrar soluciones que actúan sobre todo el sistema [20].❖ Pueden haber varios factores que están relacionados con múltiples factores, dando lugar a conexiones complejas que el pensamiento sistémico es capaz de representar [20].❖ Emplear la dinámica de sistemas ayuda a entender el sistema, lo cual permite realizar mejoras e	

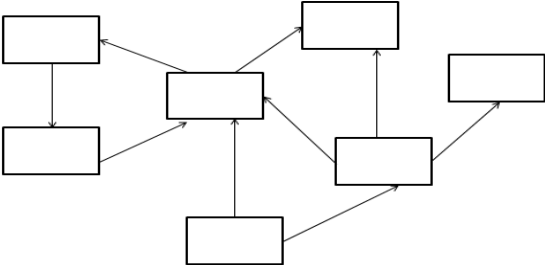
Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

<p>identificar los problemas generados [18].</p> <ul style="list-style-type: none"> ❖ A diferencia de otra herramientas de diseño lineal se admiten ciclos [19]. ❖ El pensamiento sistémico ayuda a visualizar la importancia del sistema y alertar sobre causas de posibles problemas [21]. ❖ El pensamiento sistémico ayuda a conocer con mayor detalle el sistema investigado [21].
DESVENTAJAS
<p>De los artículos revisados se pueden destacar las siguientes desventajas:</p> <ul style="list-style-type: none"> ❖ Las personas pueden pensar que ya se ha comprendido el sistema y sus interacciones, sin embargo no se tiene una visión clara su funcionamiento, lo cual lleva a tomar acciones erróneas que agudizan la situación [21]. ❖ El pensamiento sistémico suele emplear diagramas de bucles causales que conectan las variables, pero estas conexiones se realizan sin tener en cuenta los niveles de interacción, por lo cual no se logra identificar los elementos del sistema que producen el comportamiento dinámico [21].
PROCEDIMIENTO
<p>El procedimiento se hace de acuerdo al problema planteado.</p>

Tabla 4. Análisis pensamiento sistémico

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

1.5. Mapas causales (cognitive mapping- causal mapping)

MAPAS CAUSALES (COGNITIVE MAPPING- CAUSAL MAPPING)
ESQUEMA
 <p>El diagrama muestra un mapa causal con siete nodos rectangulares. Un nodo central tiene flechas que apuntan hacia los nodos superior izquierdo y superior derecho. El nodo superior izquierdo tiene una flecha que apunta hacia abajo a otro nodo. El nodo superior derecho tiene una flecha que apunta hacia abajo a un nodo. El nodo inferior izquierdo tiene una flecha que apunta hacia el nodo central. El nodo inferior central tiene una flecha que apunta hacia el nodo central. El nodo inferior derecho tiene una flecha que apunta hacia el nodo central y otra que apunta hacia un nodo más a la derecha.</p>
HERRAMIENTAS SOFTWARE
<p>Las herramientas software recomendadas para realizar los mapas causales son nombradas a continuación:</p> <ul style="list-style-type: none">❖ IHMC C-Map Herramientas COE desarrollado por el Instituto para la Cognición Humana Máquina (IHMC) [22] .❖ C-Map Tools puede ser utilizado eficazmente para representar una idea o un concepto en un formato gráfico [22].❖ www.inspiration.com [10].
DESCRIPCIÓN
<p>Los mapas cognitivos son modelos cualitativos y tienen una representación gráfica, en donde las variables son dibujadas como nodos y las relaciones entre las variables se consideran enlaces. Cada enlace es representado con flechas que muestran la dirección causal [23].</p> <p>En [24] los mapas cognitivos son empleados para mejorar la calidad de modelos conceptuales, que son el pilar en la ingeniería de requisitos.</p> <p>En [25] los mapas cognitivos han sido utilizados como una herramienta de comunicación en el desarrollo de sistemas de información.</p>
VENTAJAS
<p>De los artículos revisados se pueden destacar las siguientes ventajas:</p> <ul style="list-style-type: none">❖ Los mapas causales ayudan a identificar los problemas potenciales, lo cual facilita la toma de decisiones [23] [24].❖ Los mapas causales son herramientas útiles para modelar las complejas relaciones entre las variables [26].❖ Capacidad para permitir los procesos de retroalimentación [26].❖ Es posible modelar sistemas en los que la información científica es limitada, pero los expertos y/o conocimiento local está disponible [26].

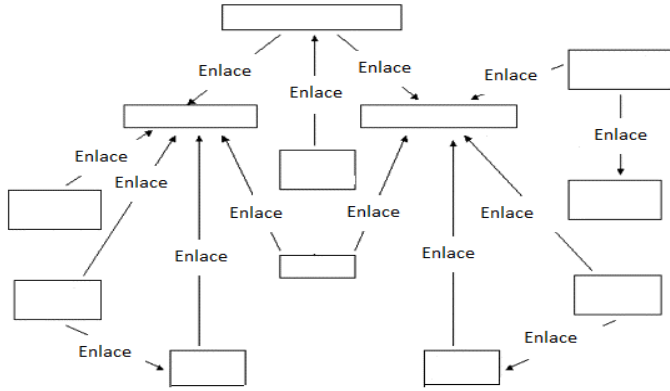
Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

<ul style="list-style-type: none"> ❖ Un mapa cognitivo es fácil y rápido de hacer [26]. ❖ Se alcanza resultados similares con tamaños de muestra más bajos en comparación con otras técnicas [26]. ❖ son herramientas eficaces para obtener la representación cognitiva humana, que juega un rol importante en la comprensión de los problemas humanos [24]. ❖ Los mapas cognitivos permiten enfocar la atención en disparadores de memoria [24]. ❖ Los mapas cognitivos pueden ayudar a resaltar las prioridades y los factores clave [24]. ❖ Los mapas cognitivos pueden suministrar información faltante [24]. ❖ Los mapas cognitivos pueden revelar las lagunas en la información o el razonamiento que necesitan una atención más directa [24]. ❖ Pueden representar datos subjetivos de manera más significativa que otros modelos y así tener utilidad para los investigadores interesados en el conocimiento subjetivo [24]
DESVENTAJAS
<p>De los artículos revisados se pueden destacar las siguientes desventajas:</p> <ul style="list-style-type: none"> ❖ El conocimiento, la ignorancia, los malentendidos y prejuicios de los entrevistados están codificados en los mapas [26]. ❖ Falta de un concepto de tiempo; es decir, no pueden modelar el comportamiento transitorio [26]. ❖ No se puede obtener datos estadísticos [26]. ❖ No pueden hacer frente a la concurrencia de múltiples causas como la expresada por condiciones ("y") [26]. ❖ Mapear los procesos cognitivos puede ser un proceso bastante tedioso [27]. ❖ El análisis puede llegar a ser extremadamente complejo si se incluyen los procesos de pensamiento de muchas personas [27].
PROCEDIMIENTO
<p>No existe un procedimiento definido para la construcción de mapas cognitivos [28]. Sin embargo se puede hacer uso de técnica como entrevistas y cuestionarios, a fin de identificar las relaciones entre los conceptos [23]. Por ejemplo en [28] realizan mapas cognitivos a partir de entrevistas grabadas, de donde se extraen conceptos y se clasifican de acuerdo a cuatro categorías 1) causa, 2) efecto, 3) valores que están en juego, 4) Gestión del riesgo. Las causas se traducen en el mapa cognitivo en nodos de incertidumbre y representan los conceptos que contienen la fuente de incertidumbre. Los efectos se convierten en nodos de efectos y representan las consecuencias de eventos inciertos. Los nodos de valor representan las consecuencias más importantes y de mayor impacto. Una vez los conceptos son identificados y clasificados, se procede a establecer las relaciones, vinculando los conceptos.</p>

Tabla 5. Análisis mapas causales

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

1.6. Mapa conceptual (concept mapping)

MAPA CONCEPTUAL (CONCEPT MAPPING)
ESQUEMA

HERRAMIENTAS SOFTWARE
Las herramientas software recomendadas para realizar los mapas conceptuales son nombradas a continuación: <ul style="list-style-type: none">❖ IHMC C-Map Herramientas COE desarrollado por el Instituto para la Cognición Humana Máquina (IHM) [22] .❖ C-Map Tools puede ser utilizado eficazmente para representar una idea o un concepto en un formato gráfico [22].❖ www.inspiration.com [10].
DESCRIPCIÓN
<p>Esta técnica está basada en las teorías de Ausubel. Un mapa conceptual es una representación gráfica donde se representa los conceptos y las relaciones, mediante nodos y enlaces respectivamente [24]. Es posible representar los pensamientos y las ideas en forma de mapas o gráficas [29].</p> <p>En [22] se emplea mapas conceptuales para llenar el vacío semántico en el desarrollo del software, para ello se usa un ejemplo de un sistema de información en salud.</p> <p>Esta técnica se enfoca en enlaces semánticos ricos entre los distintos conceptos [24].</p>
VENTAJAS
De los artículos revisados se pueden destacar las siguientes ventajas: <ul style="list-style-type: none">❖ Los mapas conceptuales permiten integrar información cualitativa desde diferentes perspectivas, lo cual contribuye en la interpretación de la información [29].❖ Los mapas conceptuales ayudan a organizar los pensamientos individuales, y representan el conocimiento [30].❖ Es fácil encontrar las relaciones entre varios temas [30] .

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

❖ Los mapas conceptuales se pueden leer de forma rápida [30].
DESVENTAJAS
De los artículos revisados se pueden destacar las siguientes ventajas:
❖ Los mapas conceptuales pueden sobrecargarse con exceso de información en el contenido y/o conexiones, por lo cual es difícil su interpretación [30].
PROCEDIMIENTO
A continuación se describe el procedimiento para elaborar mapas conceptuales, según Trochim; propuesta hecha en [29]
<p>Paso 1 Preparación: la etapa de preparación es considerada como la más importante, ya que las decisiones tomadas en este paso tienen impacto sobre todo el proceso. Para tomar decisiones se emplea la técnica lluvia de ideas, en donde se establece previamente los objetivos y resultados que se quieren obtener. Se seleccionan los participantes de acuerdo a los objetivos y resultados establecidos previamente. Se recomienda seleccionar varios participantes, con el objeto de recoger diferentes puntos de vista.</p> <p>Paso 2 Generación de declaraciones: Se recomienda intercambiar ideas entre las diferentes personas, con el propósito de generar las declaraciones. Éstas deben ser formuladas de forma clara. También se puede emplear la técnica de lluvia de ideas para generarlas.</p> <p>Paso 3 Estructuración de los estados: Para estructurar los estados es necesario realizar dos tareas: Evaluación y clasificación. Cada declaración tiene un número y se imprime en una ficha de cartulina de forma exclusiva. Un juego completo de tarjetas se da a cada participante para la estructuración individual. Para la tarea de clasificación, a cada participante se le pide que califique cada declaración en una escala de respuesta tipo Likert¹ de cinco puntos. Se ordena las declaraciones de menor importancia a mayor importancia, formando una pila. Para la tarea de clasificación cada participante debe agrupar sus fichas de acuerdo a su contenido o significado.</p> <p>Paso 4 Representación de los estados en forma de mapa conceptual: La calificación individual y clasificación de los datos de los participantes se introducen en el ordenador (Se emplea un software para mapas mentales, en [29] se recomienda utilizar Ariadne). El software analiza los datos de estructuración cuantitativamente, utilizando análisis de componentes principales, seguido por análisis de conglomerados. Las dos primeras dimensiones de cada estado se representan gráficamente en un mapa de puntos. Si hay poca distancia entre dos estados, significa que estos han sido clasificados en la misma pila, lo que implica que son conceptualmente similares de alguna manera.</p> <p>Paso 5 Interpretación: Es necesario realizar una sesión de interpretación, con el fin de observar los resultados obtenidos en la clasificación y evaluación de los conceptos.</p> <p>Paso 6 Utilización: Este paso se refiere a la finalidad del mapa conceptual.</p>

Tabla 6. Análisis mapas conceptuales

¹ Este tipo de escala mide las actitudes y los comportamientos, utilizando opciones de respuesta que van de un extremo a otro, por ejemplo poco importante, importante, muy importante, extremadamente importante.

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

1.7. Modelo de ecuaciones estructurales

MODELO DE ECUACIONES ESTRUCTURALES (STRUCTURAL EQUATION MODEL)
ESQUEMA
HERRAMIENTAS SOFTWARE
<ul style="list-style-type: none"> ❖ Linear Structural Relations [31]. ❖ AMOS (Analysis of Moment Structures; Arbuckle, 1997) [31] [32] [33] . ❖ EQS [32]. ❖ LISREL [32] [34].
DESCRIPCIÓN
<p>Los modelos de ecuaciones estructurales (SEM, abreviatura de Structural Equation Models) son una técnica de análisis estadístico multivariante², empleada para contrastar modelos que proponen relaciones causales entre las variables. La estructura teórica del modelo permite proponer estructuras causales entre las variables de manera que unas variables causen un efecto sobre otras variables que, a su vez, pueden trasladar estos efectos a otras variables [31].</p> <p>En el campo de ingeniería del software, el modelo de ecuaciones estructurales ha sido empleado para predecir debilidades específicas de las organizaciones en los procesos software [19].</p>
VENTAJAS
<p>De los artículos revisados se pueden destacar las siguientes ventajas:</p> <ul style="list-style-type: none"> ❖ Permiten proponer el tipo y dirección de las relaciones que se espera encontrar entre las muchas variables [31]. ❖ Es posible representar el efecto causal entre las variables [31]. ❖ En comparación con otros métodos de análisis multivariante se pueden mostrar las siguientes ventajas: 1) Estimación múltiple, 2) representa conceptos que no hay sido observados entre las relaciones, 3) Considera las medidas de error en las estimaciones, 4) define un modelo que explica todo un conjunto de relaciones [32]. ❖ Posee la capacidad de representar conceptos abstractos entre las muchas relaciones [33].

² Este método determina la contribución de varios factores en un evento o resultado

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

DESVENTAJAS
<p>De los artículos revisados se pueden destacar las siguientes ventajas:</p> <ul style="list-style-type: none">❖ La existencia de una correlación estadística o asociación no prueba la causalidad [32], esto puede generar confusiones, puesto que se podría afirmar que existe una relación causal sin haberla.❖ Los modelos causales son susceptibles de ser estadísticamente rechazados si se contradicen con los datos establecidos previamente en el modelo. No se debe olvidar que el valor de esta técnica es especificar complejas relaciones entre variables a-priori y luego evaluar cuantas de estas relaciones son coherentes con los datos empíricos [34].
PROCEDIMIENTO
<p>Según el trabajo realizado por María Ángeles García Veiga [35], para construir un modelo de ecuaciones estructurales se debe seguir los siguientes pasos:</p> <ol style="list-style-type: none">1. Identificar las variable observables que formaran parte del modelo y definir cómo serán medidas (generalmente obtenidas mediante escalas o cuestionarios)2. Describir un modelo teórico, donde se especifique las relaciones que se desea encontrar entre las variables. En caso de que la variable no sea directamente observable, se debe mostrar el proceso a seguir con este tipo de variables.3. Definir una matriz de covarianza³.4. Identificar los parámetros que hacen parte del modelo, para luego valorar su ajuste, si las estimaciones obtenidas no son coherentes con los datos observados⁴, el modelo debe ser corregido.5. Especificar el modelo estructural, el cual describe las relaciones causales entre las variables latentes⁵.6. Estimar los parámetros, para lo cual se debe definir los métodos de estimación a emplear. Algunos métodos son: verosimilitud (ML), mínimos cuadrados ponderados (WLS) y mínimos cuadrados generalizados (GLS).7. Identificación: el modelo estará identificado, si los parámetros del modelo pueden estimarse a partir de los elementos de la matriz de covarianzas de las variables observables.8. Finalmente se debe interpretar el modelo obtenido, a fin de contrastar los datos con el modelo propuesto.

Tabla 7. Análisis Modelo de ecuaciones estructurales

³ Grado de variación de una variable.

⁴ La información obtenida mediante escalas, cuestionarios u otros.

⁵ Es la característica que se va a medir, pero que no se puede observar.

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

1.8. Modelos gráficos (graphic model)

MODELOS GRÁFICOS (GRAPHIC MODEL)
ESQUEMA
<pre> graph TD A(()) --> B(()) A --> C(()) A --> D(()) B --> C C --> E(()) D --> E D --> F(()) E --> F </pre>
HERRAMIENTAS SOFTWARE
<p>En internet se pueden encontrar programas que permiten realizar los modelos gráficos, a continuación se menciona algunos de ellos.</p> <ul style="list-style-type: none"> ❖ Microsoft Visio ❖ MindMap
DESCRIPCIÓN
<p>Es una técnica que une la teoría de grafos y la teoría de probabilidad, gracias a su formalismo es posible modelar la estadística multivalente [36]. Consiste en una colección distribución de probabilidades que se organizan de acuerdo a una estructura de grafo [36].</p> <p>En el campo de la ingeniería del software los modelos gráficos han sido empleados para predecir debilidades específicas de las organizaciones en los procesos software [19].</p>
VENTAJAS
<p>De los artículos revisados se pueden destacar las siguientes ventajas:</p> <ul style="list-style-type: none"> ❖ Permite la captura de dependencias complejas entre variables aleatorias y la construcción de modelos estadísticos multivariantes a gran escala [36]. ❖ Permite estudiar un gran número de variables que interactúan [36].
DESVENTAJAS
<p>De los artículos revisados se pueden destacar las siguientes desventajas:</p> <ul style="list-style-type: none"> ❖ Cuando se construye un grafo de gran tamaño, puede ser difícil visualizar e interpretar la representación del gráfico [36].
PROCEDIMIENTO
<p>En la revisión realizada no se encontró un procedimiento específico para realizar modelos gráficos.</p>

Tabla 8. Análisis modelos gráfico

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

1.9. Análisis de trayectoria o de ruta

ANÁLISIS DE TRAYECTORIA O DE RUTA (CAUSAL PATH ANALYSIS)	
ESQUEMA	
<pre> graph TD X1((X1)) -- P13 --> X3((X3)) X1 -- P12 --> X2((X2)) X1 -- P15 --> X5((X5)) X2 -- P24 --> X4((X4)) X3 -- P35 --> X5 X3 -- P36 --> X6((X6)) X4 -- P45 --> X5 X4 -- P46 --> X6 X5 -- P56 --> X6 </pre>	
Donde P indica peso	
HERRAMIENTAS SOFTWARE	
<p>En internet se pueden encontrar programas que permiten realizar gráficos como los mostrados en el esquema.</p> <ul style="list-style-type: none"> ❖ Microsoft Visio ❖ MindMap 	
DESCRIPCIÓN	
<p>Análisis de trayectoria es uno de los métodos más poderosos para explorar la causalidad [37]. Esta técnica fue desarrollada y descrita por Simon y Blalock entre otros. Es una herramienta basada en mapas cognitivos, con el propósito de identificar la relación causa-efecto entre las variables [19].</p> <p>Según [37], la técnica de análisis de causa raíz es apropiada para realizar estudios en el campo de la ingeniería del software.</p>	
VENTAJAS	
<p>De los artículos revisados se pueden destacar las siguientes ventajas:</p> <ul style="list-style-type: none"> ❖ La técnica de análisis de trayectoria no exige supuestos sobre la naturaleza de los datos o tipos de distribución estadística, necesarios en otras técnicas, como regresión múltiple y ecuaciones simultáneas [37]. ❖ Utiliza como entrada los resultados de los procedimientos de correlación no paramétricos contenidos en paquetes estadísticos [37]. 	
DESVENTAJAS	
<p>De los artículos revisados se pueden destacar las siguientes desventajas:</p> <ul style="list-style-type: none"> ❖ La técnica funciona bajo el supuesto de que solo existen relaciones causales simples y unidireccionales entre dos variables [37]. 	
PROCEDIMIENTO	
<p>Según el trabajo realizado en [37], análisis de trayectoria implica los siguientes pasos:</p> <ol style="list-style-type: none"> 1. Definir una hipótesis como un sistema recursivo causal: Definir una escala para medir las variables, bajo el supuesto de que solo existen relaciones causales simples y unidireccionales entre 	

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

dos variables. Además se debe organizar un sistema recursivo.

2. Realizar recolección de datos: Los datos pueden ser recolectados a través de una variedad de procedimientos, tales como entrevistas, encuestas y estudios de campo.

3. Calcular coeficientes de correlación parcial y pesos Beta: Las entradas para el análisis de ruta causal se pueden obtener a partir de los resultados estándar de paquetes estadísticos como SPSS⁶.

4. Dibujar el diagrama de ruta causal con los resultados: En este paso se puede quitar relaciones que se consideran inadecuadas, asimismo es posible agregar relaciones que no se habían planteado, pero que son justificables.

5. Calcular y representar en una tabla los efectos directos e indirectos: Se identifican los efectos y se calculan.

Tabla 9. Análisis de trayectoria o de ruta

⁶ Es un programa estadístico informático que permite crear modelos con mayor precisión que mediante técnicas estadísticas multivariadas estándar.

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

1.10. Análisis de gráfico radial

ANÁLISIS DE GRAFICO RADIAL (RADIAL ANALYSIS CHART)	
ESQUEMA	
Imagen tomada de [38].	
HERRAMIENTAS SOFTWARE	
En internet se pueden encontrar programas que permiten realizar gráficos como los mostrados en el esquema.	
<ul style="list-style-type: none"> ❖ Power Point ❖ Apache OpenOffice Draw 	
DESCRIPCIÓN	
Dentro de la ingeniería del software, esta técnica es una extensión de los diagramas radiales, en donde cada banda radial representa un elemento al cual puede atribuirse un defecto [19].	
En [39] el análisis de gráficos radiales ha sido utilizado en la ingeniería del software como herramienta en el análisis causal de defectos.	
VENTAJAS	
De los artículos revisados se pueden destacar las siguientes ventajas:	
<ul style="list-style-type: none"> ❖ Permite observar gráficamente los resultados obtenidos, lo cual facilita la interpretación a la persona encargada de realizar el estudio [39]. ❖ El análisis de mapas radiales ayuda a identificar la etapa del software en la que se ha introducido el/los defecto(s), lo cual permite identificar puntos de mejora [39]. 	
DESVENTAJAS	
En los artículos revisados no se menciona desventajas para esta técnica.	
PROCEDIMIENTO	
A continuación se describe el procedimiento para realizar gráficos radiales, según las adaptaciones realizadas en [39], donde se trabaja con análisis causal de defectos.	
1. Seleccionar las disciplinas a analizar en el grafico radial (requisitos, arquitectura, diseño,	

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

implementación, pruebas).

2. Por cada una de las disciplinas seleccionadas se dibuja un banda radial
3. Dentro del gráfico se define el atributo defectos, el cual representa el número de defectos por cada banda, también se define el número de causas de los defectos (Esto mediante una líneas verticales).
4. Se traza una barra horizontal que sale de cada una de las disciplinas, la cual representa causas de elementos que surgen en el desarrollo del software.
5. Se traza una línea horizontal que representan el número total de defectos y otra línea horizontal que representa las causas de los defectos. En estas líneas se encuentra el total de defectos encontrados por cada disciplina y el número total de causas en cada disciplina, tal y como se muestra a continuación.

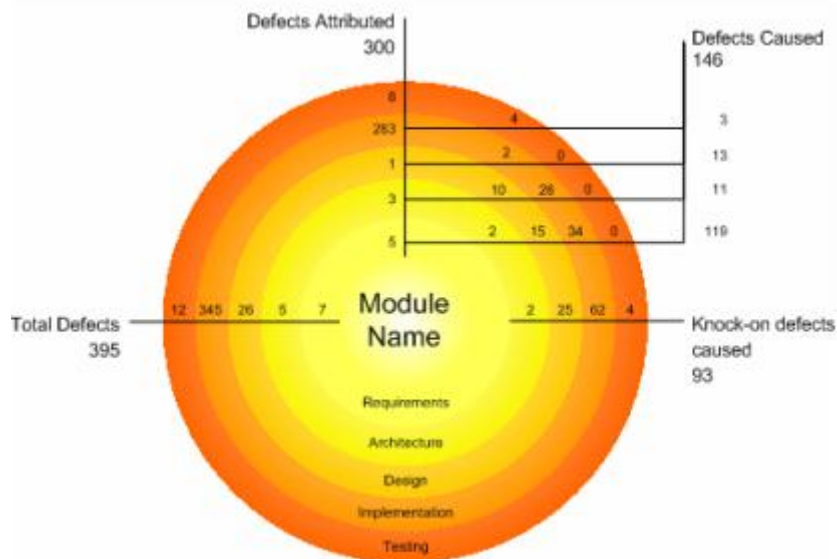
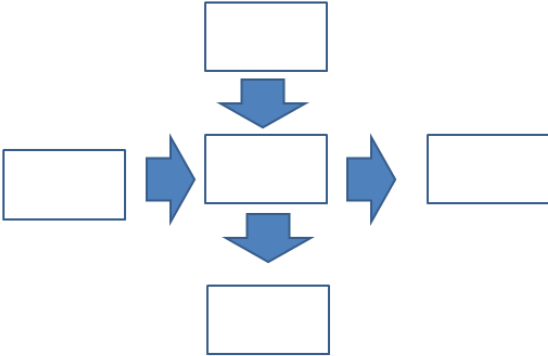


Tabla 10. Análisis de gráfico radial

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

1.11. Modelo basado en reglas

MODELO BASADO EN REGLAS (RULE-BASED MODELS)
ESQUEMA
 <pre>graph TD; A[] --> B[]; C[] --> B; D[] --> B; B --> E[]; B --> F[];</pre>
HERRAMIENTAS SOFTWARE
Programa prolog [40].
DESCRIPCIÓN
<p>El modelado basado en reglas es la forma más común de un modelo computacional [41]. Se caracteriza por un conjunto de reglas que describen una situación y producen una salida. Estas reglas codifican el conocimiento sobre los fenómenos que se están modelando [41].</p> <p>En el campo de la ingeniería del software los modelos basados en reglas han tenido las siguientes aplicaciones:</p> <ul style="list-style-type: none">❖ Han sido aplicados como parte de un sistema de predicción de defectos multi-modelo [19] [41].❖ En [42] se ha desarrollado un sistema software basado en reglas, con el fin de apoyar el proceso de captura de requerimientos en las organizaciones.❖ Los modelos basados en reglas han sido objeto de estudio en la ingeniería de datos y en el desarrollo de sistemas para la toma de decisiones [43].❖ En [40] se realiza un caso de estudio, en el cual se emplea el modelo basado en reglas para el desarrollo software.
VENTAJAS
<p>De los artículos revisados se pueden destacar las siguientes ventajas:</p> <ul style="list-style-type: none">❖ Es posible extraer el conocimiento a partir de datos, este conocimiento es representado en reglas de forma que puede ser leído y entendido por personas [41].

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

<ul style="list-style-type: none">❖ Estos modelos contribuyen en el conocimiento humano mediante las relaciones representadas [41].
DESVENTAJAS
<p>De los artículos revisados se pueden destacar las siguientes desventajas:</p> <ul style="list-style-type: none">❖ Se pueden generar reglas que no son consistentes, provocando resultados erróneos [41].❖ Se puede generar dudas sobre los resultados obtenidos en la predicción, ya que existe la posibilidad de que un dato determinado pertenezca a una categoría diferente a la mostrada en los resultados [41].❖ Según lo planteado en [42] es necesario tener una gran cantidad de datos, para extraer las reglas, ya que si no se hace esto, el modelo podría arrojar resultados con una alta tasa de error o no se podría obtener predicciones.
PROCEDIMIENTO
<p>A continuación se describe el procedimiento para realizar el modelo basado en reglas, según [41].</p> <ol style="list-style-type: none">1. Selección del atributo de resultado2. Selección de variables de predicción.3. Recopilación de datos4. Montaje del modelo5. Validación del modelo6. Actualizaciones del modelo y modificaciones.

Tabla 11. Modelo basado en reglas

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

1.12. Teoría relacional para mapas causales

TEORÍA RELACIONAL PARA MAPAS CAUSALES (RELATIONAL THEORY FOR CAUSAL MAPS)
ESQUEMA
HERRAMIENTAS SOFTWARE
<p>En internet se pueden encontrar programas que permiten realizar gráficos como los mostrados en el esquema.</p> <ul style="list-style-type: none"> ❖ Microsoft Visio ❖ MindMap
DESCRIPCIÓN
<p>Es un grafo dirigido que representa afirmaciones individuales sobre ciertas creencias respecto a un entorno [44].</p> <p>En la ingeniería del software la teoría relacional para mapas causales es usada en proyectos de inteligencia artificial [19].</p>
VENTAJAS
<p>De los artículos revisados se pueden destacar las siguientes ventajas:</p> <ul style="list-style-type: none"> ❖ De acuerdo al trabajo realizado en [44], el modelo causal planteado tiene en cuenta la relación de causalidad no reversible. Por ejemplo el licor causa enfermedad en el hígado, lo cual implica causalidad no reversible, ya que por dejar de beber no se va a curar el hígado.
DESVENTAJAS
<p>De los artículos revisados se pueden destacar las siguientes desventajas:</p> <ul style="list-style-type: none"> ❖ Es aplicable solo en grafos sin ciclos [19].
PROCEDIMIENTO
<p>A continuación se describe la teoría relacional para mapas causales, según [44]:</p> <p>Se realiza un gráfico que consta de vértices y flechas (entre los vértices), donde cada vértice representa un concepto y las flechas representan la relación causal entre los conceptos. Es decir se representa el efecto que tiene una variable sobre otra.</p>

Tabla 12. Análisis – teoría relacional para mapas causales

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

1.13. Diagrama de dispersión o de correlación

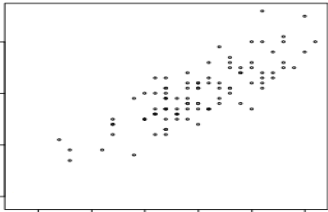
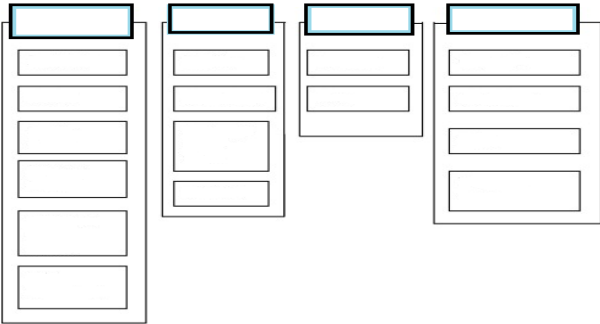
DIAGRAMA DE DISPERSIÓN O DE CORRELACIÓN
ESQUEMA

HERRAMIENTAS SOFTWARE
<p>En internet se pueden encontrar programas que permiten realizar gráficos como los mostrados en el esquema.</p> <ul style="list-style-type: none"> ❖ Microsoft-Excel ❖ ChartGo ❖ Chart ❖ Apache OpenOffice Calc ❖ LibreOffice Calc
DESCRIPCIÓN
<p>El diagrama de dispersión es una herramienta que contribuye en el estudio de posibles relaciones entre las variables [6], se construye con el propósito de mostrar la asociación existente entre éstas [45].</p>
VENTAJAS
<p>De los artículos revisados se pueden destacar las siguientes ventajas:</p> <ul style="list-style-type: none"> ❖ Permite observar gráficamente si existe relación entre dos variables [46]. ❖ El diagrama de dispersión es una herramienta útil para la solución de problemas y su eliminación [47]. ❖ Mediante la dispersión es posible observar la solidez de las relaciones [45]. ❖ Por lo general las personas llegan a una conclusión correcta, mediante el diagrama de dispersión [45]. ❖ Se muestra la naturaleza de la relación entre dos variables (lineal, curva, entre otras) [48].
DESVENTAJAS
<p>De los artículos revisados se pueden destacar las siguientes desventajas:</p> <ul style="list-style-type: none"> ❖ Las variaciones existentes en la creación del diagrama de dispersión, puede provocar la interpretación errónea en el resultado [45].
PROCEDIMIENTO
<p>Para realizar un diagrama de dispersión se coloca los valores observados en un sistema de ejes. En estos ejes se ubica las variables X y Y, junto a los pares de puntos correspondientes (x_1, y_1), (x_2, y_2), (x_3, y_3)...(x_n, y_n) . La variable dependiente es representada en el eje vertical y la variable independiente en el eje horizontal [46].</p>

Tabla 13. Análisis – Diagrama de dispersión o de correlación

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

1.14. Diagrama de afinidad

DIAGRAMA DE AFINIDAD
ESQUEMA
 El diagrama muestra cuatro columnas de ideas. Cada columna comienza con un recuadro azul que contiene una idea principal. Desde cada recuadro azul, se extienden líneas horizontales hacia abajo que conectan con recuadros blancos más pequeños, representando ideas secundarias o detalles. Las columnas están distribuidas horizontalmente y tienen diferentes alturas, lo que sugiere una organización flexible de las ideas.
HERRAMIENTAS SOFTWARE
En internet se pueden encontrar programas que permiten realizar gráficos como los mostrados en el esquema <ul style="list-style-type: none">❖ Power Point❖ Apache OpenOffice Draw
DESCRIPCIÓN
Un diagrama de afinidad es una herramienta que permite organizar un gran número de ideas de acuerdo a su relación. Esto se puede hacer mediante la lluvia de ideas, encuestas, entrevistas o resultados de retroalimentación [49]. En [50] los diagramas de afinidad han sido utilizados como un método de diseño colaborativo, para el análisis de un problema. De igual modo los diagramas de afinidad también son utilizados ampliamente en la Interacción Humano Computador (HCI).
VENTAJAS
De los artículos revisados se pueden destacar las siguientes ventajas: <ul style="list-style-type: none">❖ Los diagramas de afinidad ayuda a los equipos a agrupar y vincular sus pensamientos colectivos dentro de una estructura clara y comprensible [51].❖ Es el mejor método para tratar una gran cantidad de temas [52].❖ Es un técnica que permite obtener conocimiento, con pocos gastos económicos [53].
DESVENTAJAS
En los artículos revisados no se encuentran desventajas de esta técnica.
PROCEDIMIENTO
A continuación se describe el procedimiento planteado en [49]: <ul style="list-style-type: none">❖ Identificar el problema y expresarlo de una manera clara, concisa y fácilmente comprensible para los miembros del equipo.❖ Dar a los miembros del equipo papel y lápiz para que escriban las cuestiones relacionadas con el problema. En cada hoja de papel se debe escribir una idea, se cuenta con 10 minutos para esta actividad.❖ Coloque las hojas de papel en una superficie plana para que todos los usuarios tengan acceso a

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

ellas.

- ❖ Las hojas de papel deben ser rotadas, de modo que todos los miembros del equipo las lean.
- ❖ Se colocan las hojas de papel de acuerdo a una categoría establecida previamente, luego de llegar a un consenso entre los miembros del equipo.
- ❖ Dibujar un diagrama de afinidad y proporcionar una copia de trabajo a todos los participantes

Tabla 14. Diagrama de afinidad

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

1.15. Análisis modal de fallo y efectos

ANÁLISIS MODAL DE FALLOS Y EFECTOS (FAILURE MODE AND EFFECTS ANALYSIS)
ESQUEMA
No aplica
HERRAMIENTAS SOFTWARE
No aplica
DESCRIPCIÓN
<p>El análisis modal de fallo y efecto (FMEA, Failure Mode and Effects Analysis) es una técnica para identificar y eliminar los defectos potenciales. Ayuda a clasificar los defectos de acuerdo a su prioridad, lo cual permite tomar acciones correctivas apropiadas [54].</p> <p>Esta técnica define un número de prioridad de riesgo (RPN), que resulta de multiplicar la ocurrencia del factor del riesgo (O), la gravedad (S) y la detección (D), es decir $RPN = O \times S \times D$, donde O y S son la frecuencia y la gravedad de la falla y D representa la probabilidad de que la falla se detecte antes de que llegue al cliente [54].</p> <p>El FMEA ha sido empleado dentro de la ingeniería para identificar y priorizar los potenciales fallos en sistemas de diseño, procesos y servicios [54].</p>
VENTAJAS
<p>De los artículos revisados se pueden destacar las siguientes ventajas:</p> <ul style="list-style-type: none"> ❖ Proporciona información que ayuda a tomar decisiones de gestión de riesgos [55].
DESVENTAJAS
<p>De los artículos revisados se pueden destacar las siguientes desventajas:</p> <ul style="list-style-type: none"> ❖ La importancia relativa entre O, S y D son tratadas con el mismo peso, lo cual no puede ser cierto en las aplicaciones prácticas de FMEA [54] [55]. ❖ Las diferentes combinaciones de factores de riesgo pueden producir un valor idéntico de RPN, sin embargo pueden existir riesgos ocultos que pueden ser totalmente diferentes. Esto podría provocar que riesgos de alta prioridad pasen desapercibidos [54] [55]. ❖ Los tres factores de riesgos no pueden determinarse con precisión (O, S y D). Algunos de estos factores pueden ser imprecisos a causa de la falta de experiencia y/o datos [54]. ❖ La fórmula matemática para el cálculo de RPN es discutible y carece de una base científica completa. No hay ninguna razón de por qué O, S y D deben multiplicarse para producir la RPN [54].
PROCEDIMIENTO
<p>A continuación se describe el procedimiento planteado en [55], con la supresión de algunos paso que se consideran de poco interés para este trabajo:</p> <ul style="list-style-type: none"> ❖ Determinar los fallos de cada componente y los efectos ❖ Clasificar el nivel de peligro de cada modo de fallo ❖ Estimar la probabilidad ❖ Calcular el número de prioridad de riesgo (RPN) ❖ Determinar si la acción necesita ser tomada en función del RPN ❖ Resumir el análisis esto se puede hacer en una forma tabular

Tabla 15. Análisis modal de fallo y efectos

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

2. Definir los criterios a evaluar

Para este trabajo se ha definido un total de seis criterios, los cuales han sido seleccionados, teniendo en cuenta las características de las pequeñas organizaciones de desarrollo software, estos criterios son mostrados a continuación: 1) Referencias encontradas de su aplicación dentro de la ingeniería del software, 2) Nivel de conocimientos requeridos para el uso de la técnica, 3) Nivel de complejidad en la implementación del procedimiento, 4) Nivel de complejidad en la interpretación de esquema, 5) Herramientas de software libre disponibles y 6) nivel de conocimientos estadísticos.

A cada uno de los criterios anteriormente mencionados se le ha asignado un porcentaje en un rango de 0 a 100. El primero tiene un porcentaje del 30% porque se estima importante la relación que existe entre la herramienta y el número de trabajos publicados, donde se ha destacado su utilidad dentro de la ingeniería del software, esto podría ser un indicio de su efectividad para el presente trabajo. Para el segundo y tercer criterio se ha definido un porcentaje del 20%, debido a que se consideran relevantes teniendo en cuenta las características de las VSEs. Para los restantes se le ha establecido un porcentaje del 10%.

Estos criterios han sido seleccionados teniendo en cuenta que las pequeñas organizaciones no cuentan con los recursos suficientes para invertir en la compra de costosas licencias para programas software o en capacitación de personal. Además, es importante que las herramientas que se van a emplear en las pequeñas organizaciones sean de fácil implementación. Pino et al. [56] han extraído características de las pequeñas organizaciones, de las cuales se puede destacar poco personal para desarrollar funciones especializadas y la falta de presupuesto para contratar personas con la experiencia necesaria.

2.1. Definir escalas para los criterios seleccionados

Para seleccionar las herramientas adecuadas para el desarrollo del análisis causal se va a tomar como referente el escalamiento Likert, que es uno de los métodos más antiguo y más utilizado, el cual facilita la administración y la codificación de los datos, así como su análisis [57]. A continuación se procede a definir las escalas para cada uno de los criterios.

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

1. Referencias encontradas de su aplicación dentro de la ingeniería del software

Número de referencias	Calificación
Una o ninguna referencia	1
Dos referencias	2
Tres o cuatro referencias	3
Cinco o seis referencias	4
Siete o más referencias	5

Tabla 16. Criterio- Referencias encontradas de su aplicación dentro de la ingeniería del software

2. Nivel de conocimientos requeridos para el uso de la técnica

Nivel de conocimiento	Calificación
Conocimientos matemáticos avanzados	1
Conocimientos específicos en mejora de procesos	2
Conocimientos avanzados de ingeniería del software	3
Conocimientos básicos de ingeniería del software y/o conocimientos en el uso de programas software específicos	4
No se requiere conocimientos previos para el uso de la herramienta	5

Tabla 17. Criterio-Nivel de conocimientos requeridos para el uso de la técnica

3. Nivel de complejidad en la implementación del procedimiento

Nivel de complejidad	Calificación
Procedimiento muy complejo o sin información	1
Procedimiento complejo	2
Procedimiento medianamente complejo	3
Procedimiento sencillo	4
Procedimiento muy sencillo	5

Tabla 18. Criterio-Nivel de complejidad en la implementación del procedimiento

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

4. Nivel de complejidad en la interpretación del esquema

Nivel de complejidad	Calificación
Esquema muy complejo	1
Esquema complejo	2
Esquema medianamente complejo	3
Esquema sencillo	4
Esquema muy sencillo	5

Tabla 19. Criterio-Nivel de complejidad en la interpretación del esquema

5. Herramientas de software libre disponibles

Disponibilidad de herramientas	Calificación
No se encontró herramientas de software libre	2
Existen herramientas de software libre	5

Tabla 20. Criterio-Herramientas de software libre disponibles

6. Nivel de conocimientos estadísticos.

Nivel de conocimientos estadísticos	Calificación
Conocimientos avanzados de estadística multivariante.	1
Conocimientos de distribución	2
Conocimientos de estadística descriptiva	3
Conocimientos muy básicos de estadística descriptiva	4
No se requieren conocimientos estadísticos	5

Tabla 21. Criterio-Nivel de conocimientos estadísticos

2.2. Ponderación de los criterios

En seguida se muestra las calificaciones obtenidas al evaluar cada una de las herramientas de acuerdo a los criterios definidos anteriormente.

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

Herramienta	Referencias encontradas de su aplicación dentro de la ingeniería del software	Nivel de conocimientos requeridos para el uso de la herramienta	Nivel de complejidad en la implementación del procedimiento	Nivel de complejidad en la interpretación de esquema	Herramientas de software libre disponibles	Nivel de conocimientos estadísticos.
Diagrama de Ishikawa	5	5	5	5	5	5
Diagrama de Pareto	5	5	4	5	5	5
Mapas mentales	1	5	4	1	5	5
Pensamiento sistémico	1	3	2	2	5	5
Mapas causales	2	5	5	3	5	5
Mapa conceptual	1	5	4	3	5	5
Modelo de ecuaciones estructurales	1	1	1	3	2	1
Modelos gráficos	1	1	1	3	2	1
Análisis de trayectoria	1	1	1	3	2	1
Análisis de gráfico radial	2	5	5	5	5	5
Modelo basado en reglas	3	2	2	3	5	5
Teoría relacional para mapas causales	1	3	1	3	2	5

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

Diagrama de dispersión o de Correlación	1	1	3	4	5	3
Diagrama de afinidad	1	5	4	5	5	5
modal de fallo y efectos	4	4	3	4	2	4

Tabla 22. Calificación de las herramientas para análisis causal

La siguiente tabla muestra los resultados obtenidos al realizar los respectivos cálculos de acuerdo a los porcentajes asignados a cada criterio.

Herramienta	Referencias encontradas de su aplicación dentro de la ingeniería del software	Nivel de conocimientos requeridos para el uso de la herramienta	Nivel de complejidad en la implementación del procedimiento	Nivel de complejidad en la interpretación de esquema	Herramientas de software libre disponibles	Se requiere conocimientos estadísticos	Total
Porcentaje asignado	30%	20%	20%	10%	10%	10%	
Diagrama de Ishikawa	1,5	1	1	0,5	0,5	0,5	5
Diagrama de Pareto	1,5	1	0,8	0,5	0,5	0,5	4,8
Análisis de gráfico radial	0,6	1	1	0,5	0,5	0,5	4,1
Mapas causales	0,6	1	1	0,3	0,5	0,5	3,9

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

Diagrama de afinidad	0,3	1	0,8	0,5	0,5	0,5	3,6
Análisis modal de fallo y efecto	1,2	0,8	0,6	0,4	0,2	0,4	3,6
Mapa conceptual	0,3	1	0,8	0,3	0,5	0,5	3,4
Mapas mentales	0,3	1	0,8	0,1	0,5	0,5	3,2
Modelo basado en reglas	0,9	0,4	0,4	0,3	0,5	0,5	3
Pensamiento sistémico	0,3	0,6	0,4	0,2	0,5	0,5	2,5
Diagrama de dispersión o de Correlación	0,3	0,2	0,6	0,4	0,5	0,3	2,3
Teoría relacional para mapas causales	0,3	0,6	0,2	0,3	0,2	0,5	2,1
Modelo de ecuaciones estructurales	0,3	0,2	0,2	0,3	0,2	0,1	1,3
Modelos gráficos	0,3	0,2	0,2	0,3	0,2	0,1	1,3
Análisis de trayectoria	0,3	0,2	0,2	0,3	0,2	0,1	1,3

Tabla 23. Ponderación de los criterios para evaluar las herramientas de análisis causal

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

ANEXO 3. ESCALAS PARA EL ANÁLISIS MODAL DE FALLA Y EFECTO

1. Escala tradicional de ocurrencia (O)

Clasificación	Probabilidad de fallo	Tasa de fallo posible
10	Extremadamente alta	>1/2
9	Muy alta	1/3
8	Fracasos repetidos	1/8
7	Alto	1/20
6	Moderadamente alto	1/80
5	Moderado	1/400
4	Relativamente baja	1/2 , 000
3	Bajo	1/15 ,000
2	Remoto	1/50 , 000
1	Casi imposible	<= 1/1, 500,000

Tabla 24. Escala tradicional de ocurrencia (O)

2. Escala de clasificación de gravedad (S)

Clasificación	Efecto	Severidad del efecto
10	Peligroso sin previo aviso	Un modo de fallo que se produce sin advertencia y la consecuencia es peligrosa
9	Peligros con advertencia	Clasificación de gravedad de un modo de fallo, que se produce sin advertencia y la consecuencia es peligrosa
8	Extremo	Operación del sistema o producto se descompone sin comprometer la seguridad
7	Mayor	El funcionamiento del sistema o producto se puede continuar, pero el rendimiento del sistema o producto se ve afectado
6	Significativo	Operación del sistema o producto se continúa y el rendimiento del sistema o producto se degrada

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

5	Moderado	Rendimiento del sistema o producto se ve afectada seriamente es necesario el mantenimiento
4	Bajo	Rendimiento del sistema o producto es poco afectado y puede no ser necesario el mantenimiento
3	Menor	Se produce un efecto menor en el rendimiento del sistema y la satisfacción
2	Muy menor	Se produce un ligero efecto en el rendimiento del sistema y la satisfacción
1	Ninguno	Sin efecto

Tabla 25. Escala de clasificación de gravedad (S)

3. Escala tradicional para la detección (D)

Clasificación	Detección	Probabilidad de detección
10	Absoluta Incertidumbre	Probabilidad potencial de que ocurra el modo de falla
9	Muy remoto	La posibilidad de detectar el modo de falla es muy remota
8	Remota	La posibilidad de detectar el modo de falla es remota
7	Muy bajo	La posibilidad de detectar el potencial que ocurre de modo de fallo es muy baja
6	Baja	La posibilidad de detectar el potencial que ocurre de modo de fallo es baja
5	Moderada	La posibilidad de detectar el potencial que ocurre de modo de fallo es moderada
4	Moderadamente alta	La posibilidad de detectar el potencial que ocurre de modo de fallo es moderadamente alta
3	Alto	La posibilidad de detectar el potencial que ocurre de modo de falla es alto
2	Muy alto	La posibilidad de detectar el potencial que ocurre de modo de fallo es muy alta
1	Casi seguro	El modo de falla será detectado

Tabla 26. Escala tradicional para la detección (D)

Estas escalas han sido tomadas del trabajo realizado por Liu et al. [54]

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

ANEXO 4. TRADUCCIÓN GUÍA IBM

Etapa de apertura		
Actividad	Disparador (Triggers)	Impacto para el cliente (Impact)
<p>Revisar diseño: cuando se revisa el diseño y se compara el diseño con los requisitos establecidos.</p>	<p>Conformidad del diseño (Design Conformance): el revisor de documentos o el de revisar el código detecta el defecto, mientras compara los elementos de diseño o los segmentos de código con especificaciones realizadas en una etapa anterior.</p> <p>Ejemplos -El I código no implementó el caso para cuando no existen datos. -En una pantalla todas las fuentes están negrita, mientras que en otras pantallas no lo están.</p> <p>Logica/Flujo (Logic/Flow): El inspector utiliza el conocimiento de las prácticas y las normas básicas de programación para examinar el flujo de la lógica o los datos para asegurarse de que son correctos y completos.</p> <p>ejemplos - Memoria en esta subrutina necesita ser asignado antes de guardar estos valores. -La asignación de valor está escrito incorrectamente</p> <p>Compatibilidad con versiones anteriores (Backward Compatibility): El inspector utiliza una amplia experiencia en el producto/componente para identificar una incompatibilidad entre la función descrita por el documento de diseño o el código, y la de las versiones anteriores del mismo producto o componente.</p> <p>Ejemplo -La versión anterior tiene por defecto 0 posiciones decimales y la actual 2 posiciones.</p> <p>Documentos internos (Internal Document): hay información incorrecta, inconsistente o incompleta en la documentación existente Prólogos y comentarios del código representan algunos ejemplos de la documentación que entrarían en esta categoría. -El prólogo de subrutina A se enumeran sólo 3 parámetros cuando en realidad hay 4.</p> <p>Compatibilidad lateral (Lateral Compatibility): El inspector con experiencia de amplia base,</p>	<p>Instalabilidad (Installability): La capacidad del cliente para preparar y colocar el software en posición de uso. (No incluye Usabilidad). Ejemplos: Durante la instalación automatizada, tiene un mensaje de error diciendo que la instalación falló debido a un archivo que faltaba.</p> <p>Integridad/Seguridad (Integrity/Security): La protección de los sistemas, programas y datos de la destrucción accidental o maliciosa, alteración o divulgación. Ejemplo: Roles que tienen permiso de lectura sobre ciertos archivos pueden guardar cambios e incluso eliminar un componente.</p> <p>Rendimiento (Performance): la velocidad del software según la percepción de los clientes y los usuarios finales, en términos de su capacidad para realizar sus tareas. Ejemplo: El módulo ISGGRP00 no debe sostener el bloqueo local GRS durante tanto tiempo, porque hace</p>

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

	<p>detecta una incompatibilidad entre la función descrita por el documento de diseño o el código, y los otros sistemas, productos, servicios, componentes o módulos con los que debe interactuar.</p> <p>Ejemplo -Una aplicación grafica no permite leer un archivo gif o algún otro paquete de aplicaciones que se supone debería leerlo.</p> <p>conurrencia (Concurrency): El inspector está considerando la serialización necesaria para el control de un recurso compartido cuando se descubre el defecto. Esto incluiría la serialización de múltiples funciones, hilos, procesos o contextos del núcleo, así como la obtención y liberación de bloqueos.</p> <p>Ejemplos -La rutina A no libera el bloqueo antes de llamar a la rutina B que solicita el bloqueo. - Esta variable se ha restablecido de forma incorrecta a 0, por una rutina después de que la rutina B ya había incrementado el valor.</p> <p>Dependencia del lenguaje (Language Dependency) : El desarrollador detecta el defecto durante la comprobación de los detalles específicos del lenguaje con el cual se ha implementado un componente o una función. Normas del lenguaje, las preocupaciones de compilación y eficiencias específicas del lenguaje son ejemplos de áreas potenciales de preocupación.</p> <p>Ejemplo -El inspector está comprobando código C y ve un "=" que deberían ser "==".</p> <p>Efectos secundarios (Side Effect): El inspector utiliza una amplia experiencia o conocimiento del producto para prever algún comportamiento del sistema, producto, función o componente que puede resultar de diseño o código que se examina. Los efectos secundarios se caracterizan como resultado de uso común o configuraciones, pero fuera del alcance del componente o función con la que el diseño o código es asociado.</p> <p>Ejemplo - Mientras observa un puntero que tiene acceso a los datos por byte, el inspector se da cuenta en otra parte del código que los bytes podrían estar en orden inverso, de modo que el puntero no se puede acceder a ellos correctamente</p> <p>Situación rara (Rare Situations): El inspector utiliza una amplia experiencia o conocimiento del</p>	<p>que el otras funcionalidades no se encuentren disponibles. Después de procesar un cierto número de peticiones que debe liberar y volver a obtener el bloqueo con el fin de dar a otras unidades de trabajo la oportunidad de ejecutase.</p> <p>Mantenimiento (Maintenance): La facilidad de la aplicación de las correcciones preventivas o correctivas para el software. Ejemplo: El mantenimiento requiere una gran cantidad de esfuerzo manual.</p> <p>Facilidad de servicio (Serviceability): La capacidad para diagnosticar rápidamente las fallas, con un impacto mínimo en el cliente. Ejemplo: El software de diagnóstico muestra una cantidad de mensajes de error, en lugar de indicar dónde se produjo realmente el problema.</p> <p>Migración (Migration): La facilidad de actualizar a una versión actual, sobre todo en términos del impacto sobre los datos y las operaciones de los clientes existentes. Esto incluye la planificación de la migración, donde la falta de documentación adecuada hace difícil esta tarea. También se aplicaría en aquellos casos en que una nueva versión de un producto existente introduce cambios que afectan las interfaces externas entre el producto y las aplicaciones del</p>
--	--	---

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

	<p>producto para prever algún comportamiento del sistema que no ha sido considerado</p> <p>Ejemplo</p> <ul style="list-style-type: none"> - Requisitos afirman que una red puede manejar 250 máquinas. Un error se produce cuando un cliente tiene 1.000 máquinas. Estas no son cargas de estrés porque, ya que se ha establecido que solo 250 máquinas pueden ser alojadas 	<p>cliente.</p> <p>Ejemplo:</p> <ul style="list-style-type: none"> -Información de Co-requisito con respecto a otros productos no se puso a disposición de los clientes. - Por la migración a un nuevo nivel, las aplicaciones del cliente fallan porque la interfaz externa se ha cambiado y ya no acepta espacios en blanco. Esta compatibilidad obliga al cliente a volver a escribir 36 aplicaciones
<p>Inspección de código: cuando se examina el código o cuando se compara el código con el diseño documentado.</p>	<p>Maneja los mismos disparadores que la actividad anterior.</p>	<p>Documentación (Documentation): El grado en que la ayuda documentada para la comprensión de la estructura y los usos previstos del software son correctos y completos.</p>
<p>Pruebas unitarias: Las pruebas de caja blanca o de ejecución basado en un conocimiento detallado de los componentes internos de código.</p>	<p>Ruta sencilla (Simple Path): El caso de prueba fue motivada por el conocimiento de las ramas específicas en el código y no por el conocimiento externo de la funcionalidad.</p> <ul style="list-style-type: none"> - Trató de ejecutar el "default" de una declaración de caso, pero ya que no existía, la prueba ha fallado. <p>Ruta Complejo (Complex Path): En las pruebas de blanco / gris caja, el caso de prueba que encontró el defecto estaba ejecutando algunas combinaciones artificiales de las rutas de código. En otras palabras, el probador intentó invocar la ejecución de varias ramas bajo varias condiciones diferentes.</p> <ul style="list-style-type: none"> -ruta fracaso porque una parte de la subrutina libero memoria que posteriormente se empleó en otra parte de la subrutina. 	<p>Ejemplos:</p> <ul style="list-style-type: none"> -MSGISG015I RCAA78 no está documentada en el manual de los mensajes del sistema <p>Usabilidad (Usability): el grado en el cual el software y las ayudas publicadas permiten que el producto sea de fácil entendimiento y empleado correctamente por el usuario final.</p> <p>Ejemplo:</p> <ul style="list-style-type: none"> - En algunas situaciones, el campo de fecha no se ha rellenado. - Cuando se ejecuta varios trabajos en la prueba del sistema, el sistema se inundó de mensajes. Se desplazaban tan rápido que no se podían leer.
<p>Pruebas funcionales: ejecución de caja negra, basado en especificaciones</p>	<p>Cobertura (Coverage): durante las pruebas de caja blanca, el caso de prueba que encontró el defecto fue un intento directo para ejecutar código de una sola función, sin usar parámetros o un conjunto de parámetros.</p> <ul style="list-style-type: none"> - El probador ha intentado eliminar una ciudad de la base de datos pero no se pudo eliminar. - Código = '10 '. Este mensaje se emite de forma incorrecta debido a una condición de error que no 	

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

<p>externas de</p>	<p>existe realmente.</p> <p>Variación (variation): durante las pruebas de caja negra, el caso de prueba que encontró el defecto fue un intento directo para ejercer código para una sola función, pero utilizando una variedad de entradas y parámetros. Estos pueden incluir parámetros no válidos, valores extremos, condiciones de contorno, y combinaciones de parámetros.</p> <p>- Cuando el probador tratado de añadir un carácter más que el máximo permisible</p> <p>Secuenciación (Sequencing): Durante las pruebas de caja Negro, el caso de prueba que encontró el defecto ejecuta varias funciones en una secuencia muy específica. Este disparador sólo se elige cuando cada función se ejecuta correctamente y de forma independiente, pero falla en esta secuencia específica. También puede ser posible ejecutar una secuencia diferente con éxito.</p> <p>- El caso de prueba primero agrega un registro, y luego lo borra, y finalmente trata de agregarlo de nuevo, pero se imprime el mensaje "No se puede añadir un registro que ya existe." - La tecla "+" se pulsa dos veces, y el programa se bloquea.</p> <p>Interacción (Interaction): Durante las pruebas de caja negra, el caso de prueba que encontró el defecto inició una interacción entre dos o más cuerpos de código. Este disparador sólo se elige cuando cada función se ejecuta correctamente y se ejecuta de forma independiente, pero no en esta combinación específica.</p>	<p>Normas (Standards): El grado en que el software cumple con las normas pertinentes establecidas. Ejemplo: -El menú de comandos se presenta en la parte inferior de la pantalla en lugar de la parte superior, que es el estándar de la industria. -especificaciones del protocolo para participar en un intercambio entre sistemas heterogéneos, no se están siguiendo.</p> <p>Confiabilidad (Reliability): la capacidad del software para estar constantemente en funcionamiento, sin interrupciones no planificadas. Ejemplo: Mientras se daba la invocación al software del modem, el sistema se bloqueó y tuvo que ser reiniciado.</p> <p>Requisitos (Requirements): una expectativa del cliente, en cuanto a la capacidad que no fue conocida, entendida o priorizada como requisito para el producto o versión actual. Este valor debe ser elegido durante el desarrollo de adiciones al plan de registro. Cuando los clientes reportan que la funcionalidad encontrada no cumple con sus expectativas. Ejemplos: El cliente necesita que el software tenga la capacidad de tomar la entrada desde el teclado, el ratón o archivos planos.</p>
<p>Pruebas del sistema: Prueba de ejecución del sistema completo, en un entorno real, que requiere de todos los recursos.</p>	<p>Pruebas de carga de estrés(Workload/Stress): El sistema está funcionando en o cerca de algún límite de recursos, ya sea superior o inferior. Estos límites de recursos se pueden crear por medio de una variedad de mecanismos, incluyendo la ejecución de cargas pequeñas o grandes, corriendo algunos o muchos productos a la vez, dejando que el sistema funcionará por un período prolongado de tiempo.</p> <p>- Cuando el sistema está inactivo durante 10 minutos, se cuelga.</p> <p>Recuperación/Excepción (Recovery/Exception): El sistema está siendo probado con la intención de invocar un manejador de excepciones o algún tipo de código de recuperación. El defecto no habría surgido si alguna excepción anterior no hubiera causado la excepción o el proceso de recuperación hubiera sido invocado.</p> <p>Inicio/reinicio (Startup/Restart): el sistema o subsistema ha sido inicializado o restaurado antes de</p>	<p>Accesibilidad (Accessibility):</p>

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

	<p>apagarse, el sistema completo ha fallado o se presentan fallos en el subsistema.</p> <p>-Después de desenchufar el cable de la CPU, el software no fue capaz de ponerse en marcha de nuevo hasta que se limpiaron algunos archivos que quedaron en uno de los directorios. -Durante el proceso de reinicio automático, un sistema falla con MSGISG0151 y terminación anormal.</p> <p>Configuración del hardware (Hardware Configuration): el sistema ha sido probado para asegurar que las funciones se ejecuten correctamente en configuraciones específicas del hardware. -Encontrar un defecto al enviar cualquier trabajo a una impresora de determinada marca.</p> <p>Configuración del software (Software Configuration): el sistema ha sido probado para asegurar que las funciones se ejecuten correctamente en configuraciones específicas de software.</p> <p>-TRYJOIN no funciona en un entorno non-sysplex</p> <p>Pruebas de bloqueo (Blocked Test): El producto está funcionando bien dentro de los límites de recursos y el defecto surgió al intentar ejecutar un escenario de prueba del sistema. Este disparador se utiliza en situaciones en las que no se pudo realizar la ejecución porque hay problemas básicos que lo impiden. Este disparador no debe ser utilizado en los defectos informados por los clientes.</p> <p>-Durante la prueba del sistema, se quería comprobar la tensión de carga de trabajo mediante la impresión de 1.000 trabajos. Sin embargo, cuando se hizo clic en Imprimir, no pasó nada. No aparece la pantalla para solicitar la entrada.</p>	<p>Asegurar que el acceso exitoso a la información y el uso de tecnología de la información se proporciona a las personas que tienen discapacidades.</p> <p>Capacidad (Capability): La capacidad del software para realizar las funciones previstas, y satisfacer los requisitos CONOCIDOS, donde el cliente no se ve afectada en ninguna de las categorías anteriores. -Cuando se hace clic en Guardar, no pasó nada</p>
revisión ID	<p>Precisión: La información no describe el producto correctamente.</p> <p>Claridad: la información es confusa o difícil de entender.</p> <p>Integridad: La información necesaria no se encuentra.</p> <p>Consistencia: La información no se muestra de una manera consistente. Hay diferentes explicaciones, ortografía, redacción, etc ... para la misma pieza de información</p> <p>Organización: No se refleja la relación entre las partes y el todo.</p> <p>Recuperabilidad: La información es difícil de encontrar.</p>	

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

	<p>Estilo: La forma de expresión es inadecuada o difícil de entender.</p> <p>Orientación de la tarea: La presentación de por qué y cómo realizar una tarea es inapropiado.</p> <p>Diseño gráfico / Estética: El aspecto y el diseño de la información es inadecuada.</p>	
Revisar GUI	<p>Conformidad de diseño (Design Conformance): Los defectos encontrados durante la revisión de documentos de diseño de interfaz gráfica de usuario.</p> <p>Widget/Icon Appearance: Defectos encontrados durante la revisión de widget y la apariencia de los iconos (color, tamaño, etc)</p> <p>Texto de pantalla/Caracteres (Screen Text/Characters): Los defectos encontrados durante la revisión de texto o caracteres, por ejemplo, un botón mal etiquetado.</p> <p>Dispositivos de entrada (Input Devices): Incluye defectos encontrados durante las pruebas de varios métodos de entrada, incluyendo el ratón, teclado, pantalla táctil.</p> <p>Navegación (Navigation): Los defectos encontrados al intentar navegar dentro o entre las ventanas, menús, o componentes.</p> <p>Widget/GUI Behavior: defectos encontrados mientras se probaba el comportamiento esperado de widget e iconos (varios elementos de lista termina con énfasis de selección, botón de ayuda no funciona).</p>	

Tabla 27. Etapa de apertura ODC

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

Etapa de cierre					
Actividad	Origen (Target)= Design/Code	Tipo de defecto (Defect Type)	Calificador (Qualifier)	Fuente (Source)	Edad (Age)
<p>Revisar diseño/Inspección de código/Pruebas Unitarias/Pruebas de función/Pruebas del sistema</p>	<p>Requisitos (Requirements): con el fin de corregir el defecto, era necesario cambiar el documento de requisitos. Ejemplos: -Debe ser capaz de aislar la fuente de fracaso en un entorno abierto. -Debe proporcionar apoyo para nueva configuración de software.</p> <p>Diseño (Design): Con el fin de corregir el defecto, era necesario cambiar el documento de especificación de diseño. Ejemplos: -Interfaz con el nuevo DAsD no se incluyó en el diseño. -paginación excesiva resultará en un impacto de rendimiento.</p> <p>Código (Code): Con el fin de corregir el defecto, era necesario cambiar el código. Ejemplos: 1) La falta de comprobar los códigos de retorno. 2) asignación incorrecta.</p>	<p>Asignación/inicialización (Assignment/Initialization): Valor(s) asignado incorrectamente o no asignado en absoluto; pero tenga en cuenta que una solución que implica múltiples correcciones de asignación puede ser del tipo de algoritmo. Ejemplos: 1) la variable interna o una variable dentro de un bloque de control no tenía el valor correcto, o no tenían ningún valor en absoluto. 2) Inicialización de los parámetros 3) Restablecer el valor de una variable. 4) La variable de instancia la captura de una característica de un objeto (por ejemplo, el color de un coche) se omite. 5) Las variables de instancia que capturan el estado de un objeto no se inicializan correctamente.</p> <p>Comprobación (Checking): Los errores causados por la falta o la validación incorrecta de los parámetros o datos en las sentencias condicionales. Se podría esperar que una consecuencia de la comprobación de un valor requeriría código adicional tal como un ciclo hacer mientras. Si la comprobación o el chequeo incorrecto es el error crítico. La comprobación debe ser el tipo seleccionado. 1) Valor superior a 100 no es válida, pero el valor era inferior a 100. 2) El bucle condicional debería haber parado en la novena iteración. Pero se mantuvo hasta que el contador fue ≤ 10.</p> <p>Algoritmo/Método (Algorithm/Method): problemas de eficiencia o exactitud que afectan</p>	<p>Que falta (Missing): El defecto se debe a una omisión, por ejemplo una sentencia de asignación faltaba.</p> <p>Incorrecta (Incorrect): el defecto se debe a una comisión, por ejemplo una declaración de control usó valores incorrectos.</p> <p>extraños (Extraneous): El defecto se debe a algo que no es relevante o pertinente al documento o código (por ejemplo) hay una sección del documento de diseño que no es pertinente para el producto actual y debe ser eliminado</p>	<p>Desarrollado In-House (Developed In-House): El defecto se encuentra en una zona que fue desarrollado internamente por el propio equipo de desarrollo de la organización.</p> <p>Reutilizada de biblioteca (Reused From Library): el defecto es encontrado usando un estándar de librería reutilizable. El problema podría ser que la parte reutilizada se utiliza incorrectamente o que hay un problema dentro de la parte reutilizada en sí.</p> <p>Subcontratado (Outsourced):</p>	<p>Base: el defecto es una parte del producto que no ha sido modificado por el proyecto actual y no es parte de la reutilización de una biblioteca estándar. El defecto no se inyecta por el proyecto actual, por lo tanto era un defecto oculto.</p> <p>Nueva (New): El defecto está en una función que fue creado por y para el proyecto actual y que introduce la nueva función.</p> <p>Refundido(Rewritten): El defecto se introdujo como un resultado directo de rediseño y / o de la función de reescritura en un intento de mejorar su diseño o calidad.</p> <p>Refixed: El defecto fue introducido por la solución aportada para arreglar un defecto anterior.</p>

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

	<p>Construir/Paquete (Build/Package): Con el fin de corregir un defecto descubierto durante el proceso, era necesario introducir cambios en la construcción del proceso, librerías del sistema o con la gestión de cambios o el control de versiones. Ejemplos: -Parte no definida para construir procesos. Macros de mapeo requirieron montar programas de aplicación que faltan en una librería pública. -Módulo compilado utilizando una versión de nivel anterior de una parte del código fuente.</p> <p>Desarrollo de la Información (Information Development): El problema es con la descripción escrita contenida en guías de usuario, manuales de instalación, ayuda en línea, los mensajes de usuario Ejemplos: -Una instalación problema se experimenta porque no había</p>	<p>a la tarea y pueden fijarse por (re) implementación de un algoritmo o estructura de datos local sin la necesidad de solicitar un cambio de diseño. Problema en el procedimiento, la plantilla, o función sobrecargada que describe un servicio ofrecido por un objeto. Ejemplos: -El algoritmo para la búsqueda de una cadena de bloques de control se corrigió a utilizar una lista lineal en lugar de una lista circular. -se especifican de forma incorrecta el número y / o tipos de parámetros de un método o una operación. -Un método o una operación no se harán públicas en la especificación de una clase</p> <p>Función/Clase/Objeto (Function/Class/Object): el error debe exigir un cambio de diseño formal, ya que afecta la capacidad significativa, las interfaces de usuario final, las interfaces de productos, la interfaz con la arquitectura de hardware, o una estructura de datos global (s); El error se produjo durante la ejecución del estado y capacidades una entidad abstracta o real. Ejemplos: -Una base de datos no incluye un campo para la dirección de la calle, aunque los requisitos lo especificaban. -Una base de datos incluye un campo para el código postal de correos, pero ya era demasiado pequeño para contener códigos postales internacionales como se especifica en los requisitos. - A C ++ o la clase SmallTalk se omitió durante el diseño del sistema.</p> <p>Sincronización/Serialización</p>		<p>Un defecto es en una parte proporcionada por un proveedor.</p> <p>portado (Ported): El defecto tiene que ver con el uso de una parte que se validó para un entorno diferente.</p>	
--	---	---	--	---	--

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

	<p>documentación de usuario de los conjuntos de datos que se asignarán antes de intentar la instalación.</p> <p>Soporte de idioma nacional (National Language Support): Se hicieron las correcciones a la aplicación de las funciones del producto en otros idiomas aparte del inglés. Ejemplos: -El mensaje contiene la basura para el francés sólo -No se puede imprimir caracteres específicos alemanes</p>	<p>(Timing/Serialization): la serialización para compartir recursos ha fallado. La serialización del recurso fue incorrecta o se empleó la técnica de serialización mal. Ejemplo: - La serialización falta al hacer cambios a un bloque de control compartido</p> <p>Interfaces/Mensajes O-O(Interface/O-O Messages): Problemas de comunicación entre: 1) Módulos 2) componentes 3) los controladores de dispositivos 4) objetos 5) las funciones 1) macros 2) declaraciones de llamadas 3) bloques de control Ejemplos: - El número y / o tipos de parámetros de la mensaje-OO no se conforman con la firma del servicio solicitado. - La interfaz especifica un puntero a un número, pero la aplicación está esperando un puntero a un carácter</p> <p>Relación (Relationship): los problemas relacionados con las asociaciones entre los procedimientos, estructuras de datos y objetos. Tales asociaciones pueden ser condicionales. Ejemplos: -La estructura del código / datos en un solo lugar supone una cierta estructura de código / datos en otro. Sin consideración apropiada de su relación, el programa no se ejecutará o se ejecuta de forma incorrecta. -La relación de herencia entre dos clases es incorrecta o se encuentra mal especificada. - El límite en el número de objetos que pueden</p>			
--	---	--	--	--	--

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

		ser instanciados de una clase dada es incorrecta y causa la degradación del rendimiento del sistema.			
revisión ID	Información de Desarrollo	<p>Editorial (Editorial): Los defectos se refiere a la gramática, la ortografía, puntuación, organización, etc ... necesarios para construir la información que describe un producto. Ejemplos: -El texto de ayuda que se muestra tiene un error de ortografía.</p> <p>Técnica (Technical): Los defectos relacionados con la descripción de un producto y sus interfaces. Ejemplos: -La Guía de instalación muestra un diagrama de cableado incorrecto de conectar el módem a un ordenador.</p> <p>Navegación (Navigational): los defectos que impidan a los usuarios encontrar la información necesaria sobre un producto que incluye los errores en los mecanismos utilizados para conectar la información (es decir, hipervínculos, índices, tablas de contenidos, referencias cruzadas, estantes de libros, etc ...). ejemplo: -Cuando el usuario hace clic en un hipervínculo en la web, que recibió el mensaje "Netscape no puede encontrar el servidor ..."</p>			
Revisar GUI	Diseño/Código				

Tabla 28. Etapa de cierre ODC

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

ANEXO 5. ENTREVISTAS A LOS INVOLUCRADOS EN EL PROYECTO

1. ¿El procedimiento permitió evidenciar los defectos?

Sí___ No___

Observaciones

2. ¿El procedimiento permitió evidenciar las causas de los defectos?

Sí___ No___

Observaciones

3. ¿La técnica de Ishikawa fue de fácil uso?

Sí___ No___

Observaciones

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

4. ¿La técnica de diagramas de afinidad fue de fácil uso?

Sí ___ No___

Observaciones

5. Para el procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software, indique los siguientes aspectos:

Aspectos positivos	Aspectos a mejorar

6. ¿El procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software ha sido útil para el desarrollo del proyecto?

Sí ___ No___

Observaciones

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

7. ¿El procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software ha sido fácil de llevar a cabo?

Sí ___ No ___

Observaciones

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

ANEXO 6. ENTREVISTAS PARA EL CLIENTE

6.1. Diseño 1

1. El software cumple con todas las funcionalidades acordadas para la segunda entrega.

Sí___ No___

En caso de que la respuesta sea negativa, indique cuáles funcionalidades no fueron entregadas.

2. Está usted conforme con las funcionalidades entregadas

Sí ___ No___

En caso de que la respuesta sea negativa, indique los motivos de su inconformidad.

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

3. ¿Ha encontrado defectos en la aplicación?

Sí ___ No___

En caso de que la respuesta sea positiva, indique los defectos encontrados

Las preguntas de la 3 a la 6 están relacionadas con la interfaz de usuario.

4. Los iconos utilizados son representativos. Sí ___ No___

5. Los menús están debidamente agrupados. Sí ___ No___

6. Es fácil navegar entre las diferentes ventanas, menús o componentes. Sí ___ No___

7. El aspecto y el diseño de la información es adecuado. Sí ___ No___

8. De la lista mostrada a continuación, indique posibles problemas presentados con el equipo encargado de ejecutar el proyecto.

No tienen en cuenta las necesidades del cliente

El equipo de trabajo no respeta fechas ni plazos establecidos

El equipo de trabajo no ha validado los requisitos con los diferentes Stakeholders

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

No se tiene claro el alcance del proyecto

No hay comunicación entre el equipo de trabajo y el cliente

Otros:

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

6.2. Diseño 2

1. El software cumple con todas las funcionalidades acordadas.

Sí ___ No ___

En caso de que la respuesta sea negativa, indique cuáles funcionalidades no fueron entregadas.

2. Está usted conforme con las funcionalidades entregadas

Sí ___ No ___

En caso de que la respuesta sea negativa, indique los motivos de su inconformidad.

Las preguntas de la 3 a la 6 están relacionadas con la interfaz de usuario.

3. Los iconos utilizados son representativos. Sí ___ No ___

4. Los menús están debidamente agrupados. Sí ___ No ___

5. Es fácil navegar entre las diferentes ventanas, menús o componentes. Sí ___ No ___

6. El aspecto y el diseño de la información es adecuado. Sí ___ No ___

7. De la lista mostrada a continuación, indique posibles problemas presentados con el equipo encargado de ejecutar el proyecto.

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

- No tienen en cuenta las necesidades del cliente
- El equipo de trabajo no respeta fechas ni plazos establecidos
- Hay deficiencia en la comprensión del alcance del proyecto
- Falta de comunicación entre el equipo de trabajo y el cliente

Otros:

8. De la lista mostrada a continuación, indique posibles defectos presentados en la aplicación

- Apariencia de los iconos (la alineación de los iconos no es correcta)
- Errores ortográficos en el texto incluido en la aplicación
- defectos encontrados mientras se probaba el comportamiento esperado de widget e iconos
- Se muestra contenido en un idioma diferente al español
- Se muestran mensajes de error de compilación al navegar en la aplicación

Otros:

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

ANEXO 7. RECOLECCIÓN DE DATOS

La recolección de datos se realizó de la siguiente manera.

PRIMERA FASE

1. Actividad Detección de defectos

❖ Tarea *Recolección de defectos*

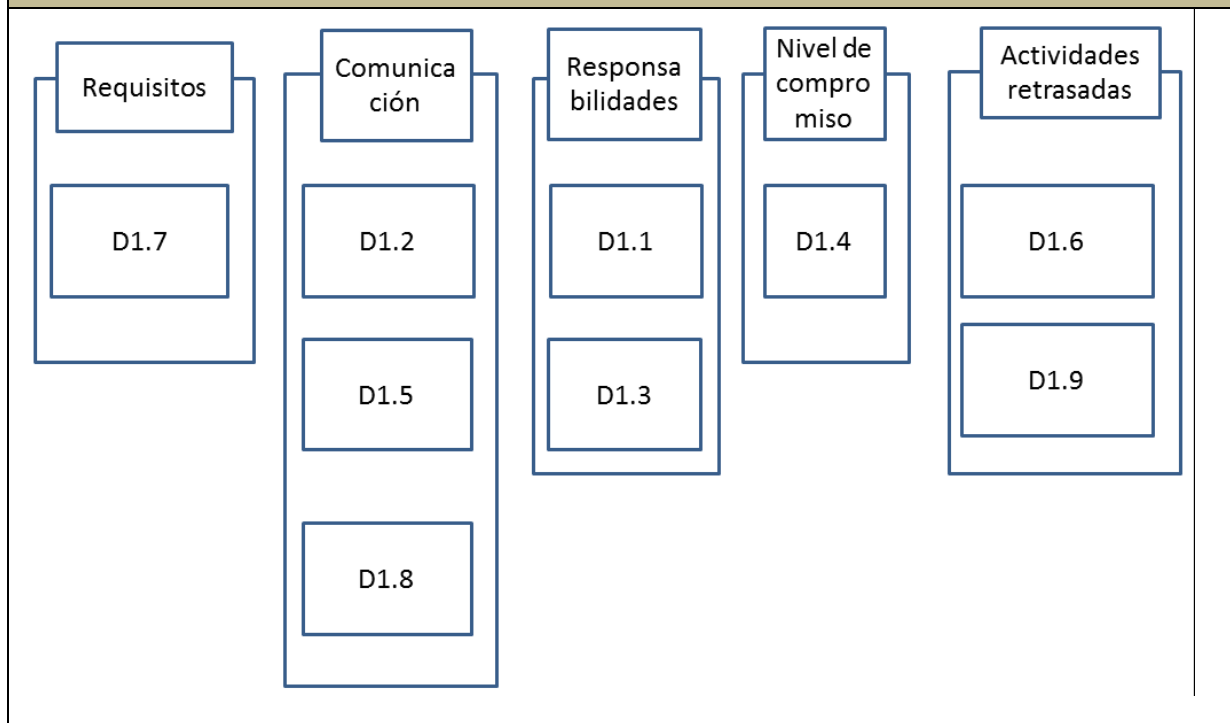
Durante esta reunión, el líder de análisis causal hizo uso de la técnica diagramas de afinidad para evidenciar los defectos.

PLANTILLA DE RECOLECCIÓN DE DEFECTOS		
PROCEDIMIENTO DE ANÁLISIS CAUSAL ENFOCADO EN PEQUEÑAS ORGANIZACIONES DE DESARROLLO SOFTWARE (PAC-DS)		
Organización	Grupo proyecto II	
Proyecto	Sistemas de gestión de actividades diarias-PAD	
Fecha: 15 de septiembre del 2014		
ID	Defecto	Cantidad de personas que lo reportaron
D1.1	Algunos de los miembros del equipo no cumplen con las tareas asignadas, ya que no se entregan a tiempo o se entregan incompletas	2
D1.2	No se revisan los mensajes colocados en la red social Facebook (Falta de comunicación en el grupo y los subgrupos)	2
D1.3	En ocasiones no se tiene claro las actividades que determinado rol debe realizar durante la semana, por lo tanto quedan personas sin trabajo asignado.	2

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

D1.4	Algunos de los miembros del equipo no asisten puntualmente a las reuniones establecidas.	4
D1.5	No hay coordinación entre los documentos del proyecto y el repositorio	2
D1.6	Falta de documentación que brinde una visión general del proyecto (fechas de entrega, responsabilidades, hitos, entre otros), como el plan del proyecto	1
D1.7	Mala interpretación de los requisitos especificados por el cliente	1
D1.8	Algunos integrantes no han logrado aún integrarse al ritmo del grupo	2
D1.9	Retrasos en la definición del proceso de desarrollo	1

DIAGRAMA DE AFINIDAD



Elaboró: Kelly Yohanna Zuñiga

Revisó: Kelly Yohanna Zuñiga

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

Fecha: 16-09-2014	Fecha: 16-09-2014
-------------------	-------------------

Tabla 29. Plantilla de recolección de defectos, fase 1

En la anterior plantilla se encuentra la lista de defectos hallados durante la reunión y los mencionados por el tutor de la materia proyecto II, el ingeniero Libardo Pantoja. También se agregan aquellos defectos mencionados por el cliente en la entrevista mostrada a continuación

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

ENTREVISTA

1. El software cumple con todas las funcionalidades acordadas.

Sí No

En caso de que la respuesta sea negativa, indique cuáles funcionalidades no fueron entregadas.

Sin embargo hay que hacerle algunos ajustes para que quede mucho mejor.
los ajustes consisten en: - Apilamiento
- Mejora del play / pause.

2. Está usted conforme con las funcionalidades entregadas

Sí No

En caso de que la respuesta sea negativa, indique los motivos de su inconformidad.

El programa tiene las funcionalidades fundamentales sin embargo es necesario mejorar.

Las preguntas de la 3 a la 6 están relacionadas con la interfaz de usuario.

3. Los iconos utilizados son representativos. Sí No

4. Los menús están debidamente agrupados. Sí No

5. Es fácil navegar entre las diferentes ventanas, menús o componentes. Sí No

6. El aspecto y el diseño de la información es adecuado. Sí No

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

7. De la lista mostrada a continuación, indique posibles problemas presentados con el equipo encargado de ejecutar el proyecto.

- No tienen en cuenta las necesidades del cliente
- El equipo de trabajo no respeta fechas ni plazos establecidos
- Hay deficiencia en la comprensión del alcance del proyecto
- Falta de comunicación entre el equipo de trabajo y el cliente

Hubo diferentes entregas y se va re entregando en el tiempo acordado.

Otros:

8. De la lista mostrada a continuación, indique posibles defectos presentados en la aplicación

- Apariencia de los iconos (la alineación de los iconos no es correcta)
- Errores ortográficos en el texto incluido en la aplicación
- Defectos encontrados mientras se probaba el comportamiento esperado de widget e iconos
- Se muestra contenido en un idioma diferente al español
- Se muestran mensajes de error de compilación al navegar en la aplicación

Otros:

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

❖ **Determinar *muestra de los defectos***

Una vez se tienen los defectos que han surgido durante la ejecución del proyecto es necesario realizar el diagrama de Pareto para determinar la muestra de los defectos. Este fue realizado de acuerdo a la Tabla 29.

Ficha del documento

Fecha	Revisión	Autor	Observaciones
16-09-2014		Kelly Yohanna Zuñiga	
Elaboró: Kelly Yohanna Zuñiga Fecha: 16-09-2014		Revisó: Kelly Yohanna Zuñiga Fecha: 16-09-2014	

Propósito

El propósito de este documento es mantener evidencias de los defectos que generan mayor impacto en el desarrollo del proyecto.

El documento contiene:

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

1. Diagrama de Pareto o análisis modal de fallo y efecto

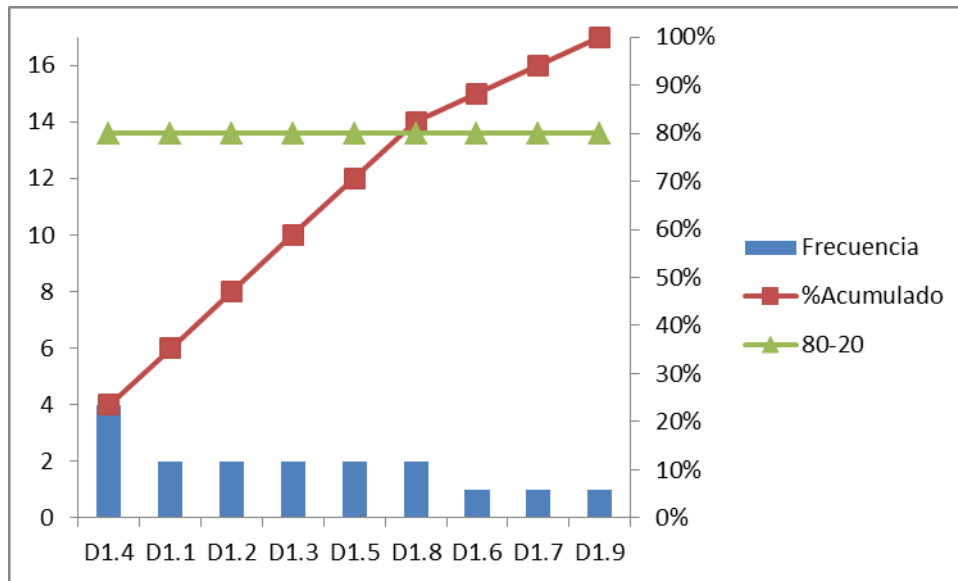


Figura 1. Diagrama de Pareto para los defectos detectados en la fase 1

Según lo mostrado en el diagrama de Pareto resultante (Figura 1), el 80% de estos defectos se deben a un 20% de causas que deben ser eliminadas. Para este propósito la línea de color rojo (% acumulado) debajo de la línea verde, muestra los defectos que deben ser abordados para su respectivo análisis.

2. Defectos de mayor impacto

Identificador	Defecto
D1.4	Algunos de los miembros del equipo no asisten puntualmente a las reuniones establecidas.
D1.1	Algunos de los miembros del equipo no cumplen con las tareas asignadas, ya que no se entregan a tiempo o se entregan incompletas
D1.2	No se revisan los mensajes colocados en la red social Facebook (Falta de comunicación en el grupo y los subgrupos)
D1.3	En ocasiones no se tiene claro las actividades que determinado rol debe realizar durante la semana, por lo tanto quedan personas sin trabajo

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

	asignado.
D1.5	No hay coordinación entre los documentos del proyecto y el repositorio
D1.8	Algunos integrantes no han logrado aún integrarse al ritmo del grupo

❖ Clasificar los defectos

Dado que los defectos evidenciados durante la primera fase están relacionados directamente con la ejecución del proyecto, se procede a realizar su clasificación y determinación de causas mediante el diagrama de Ishikawa (Tarea identificar causas).

2. Actividad Detección de causas fundamentales

❖ Tarea *Identificar causas*

Ficha del documento

Fecha	Revisión	Autor	Observaciones
17-09-2014		Kelly Yohanna Zuñiga	
Elaboró: Kelly Yohanna Zuñiga Fecha: 17-09-2014		Revisó: Kelly Yohanna Zuñiga Fecha: 17-09-2014	

Propósito

El propósito de este documento es tener información histórica que permita comparar situaciones perjudiciales presentadas previamente y evitar su aparición en proyectos futuros. Asimismo este reporte especifica el responsable de realizar los diagramas correspondientes.

El documento contiene:

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

1. Diagramas

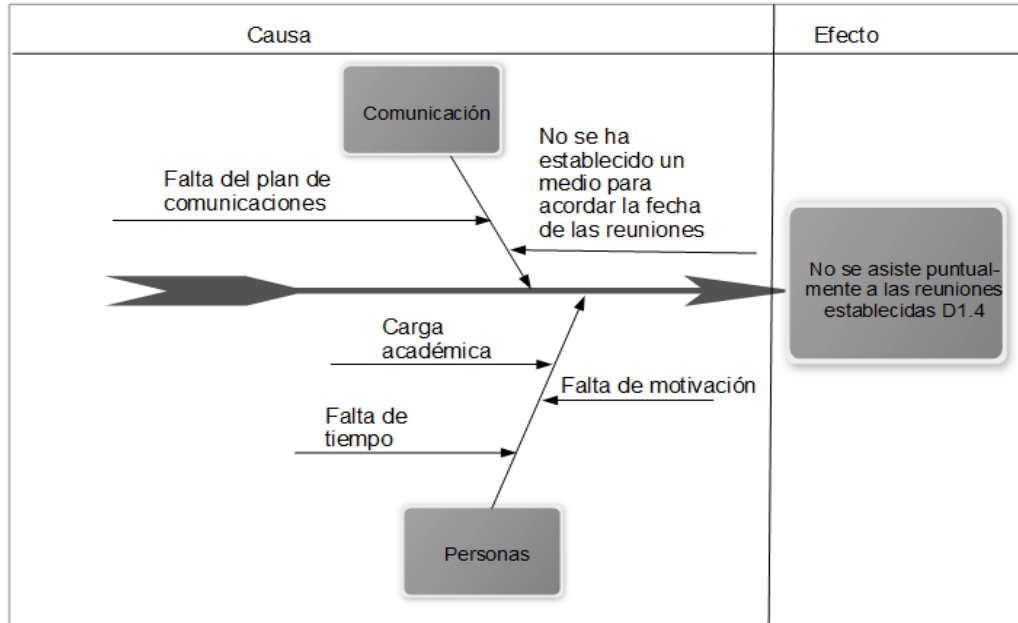


Figura 2. Diagrama de Ishikawa D1.4

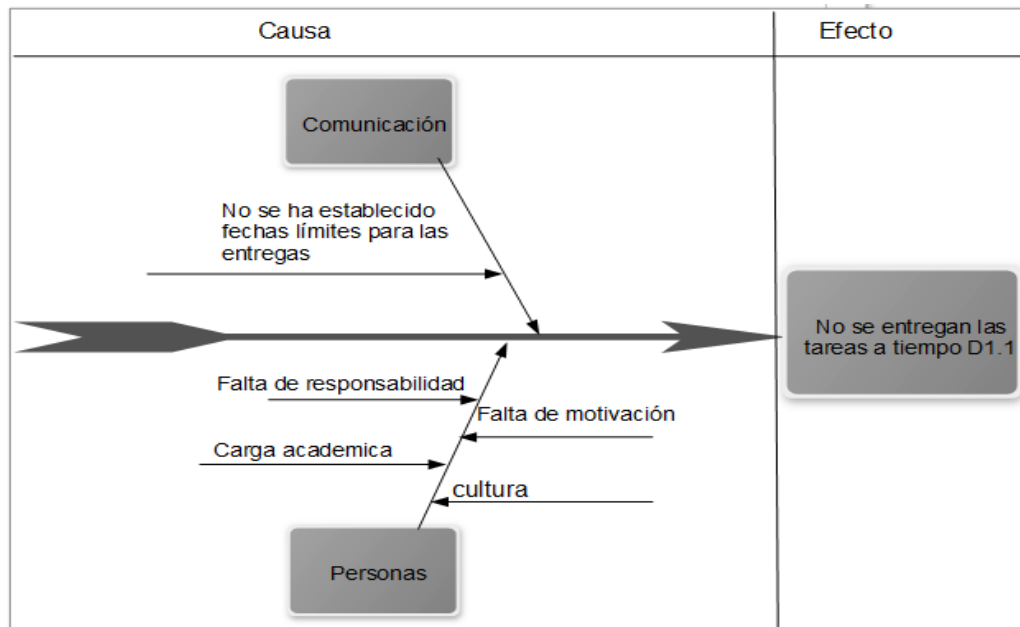


Figura 3. Diagrama de Ishikawa D1.1

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

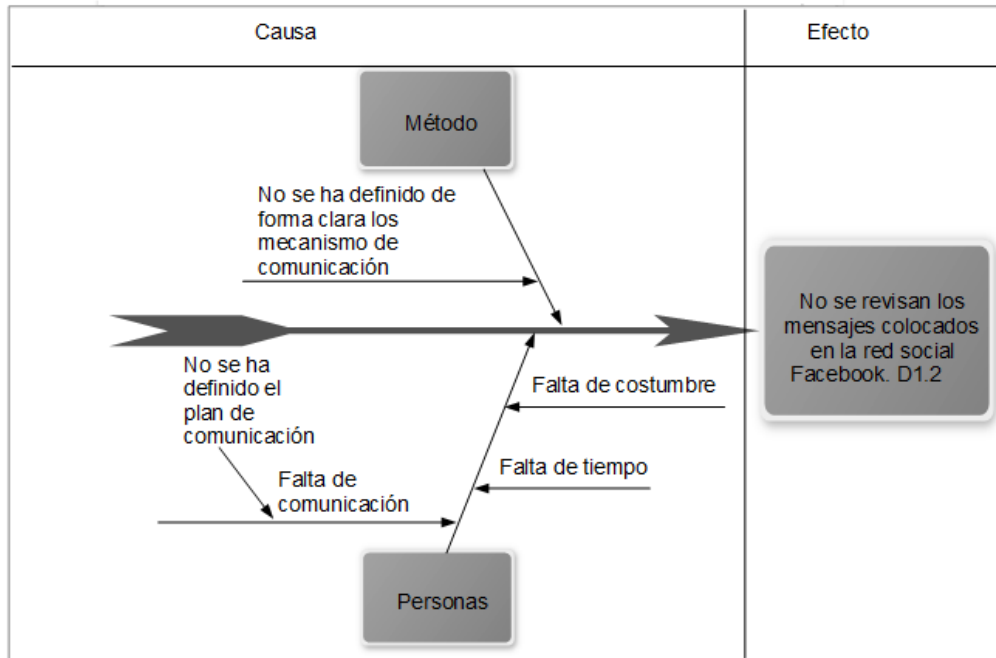


Figura 4. Diagrama de Ishikawa D1.2

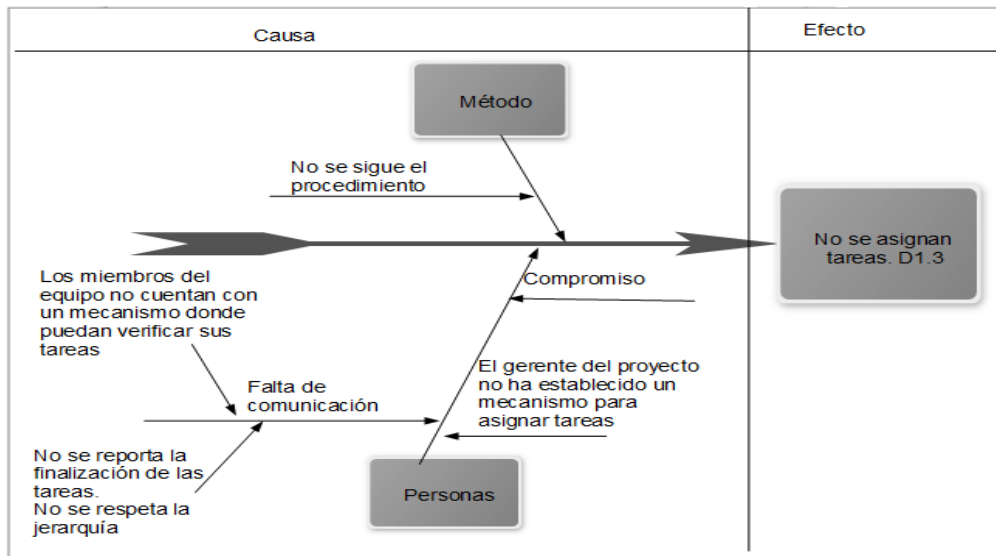


Figura 5. Diagrama de Ishikawa D1.3

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

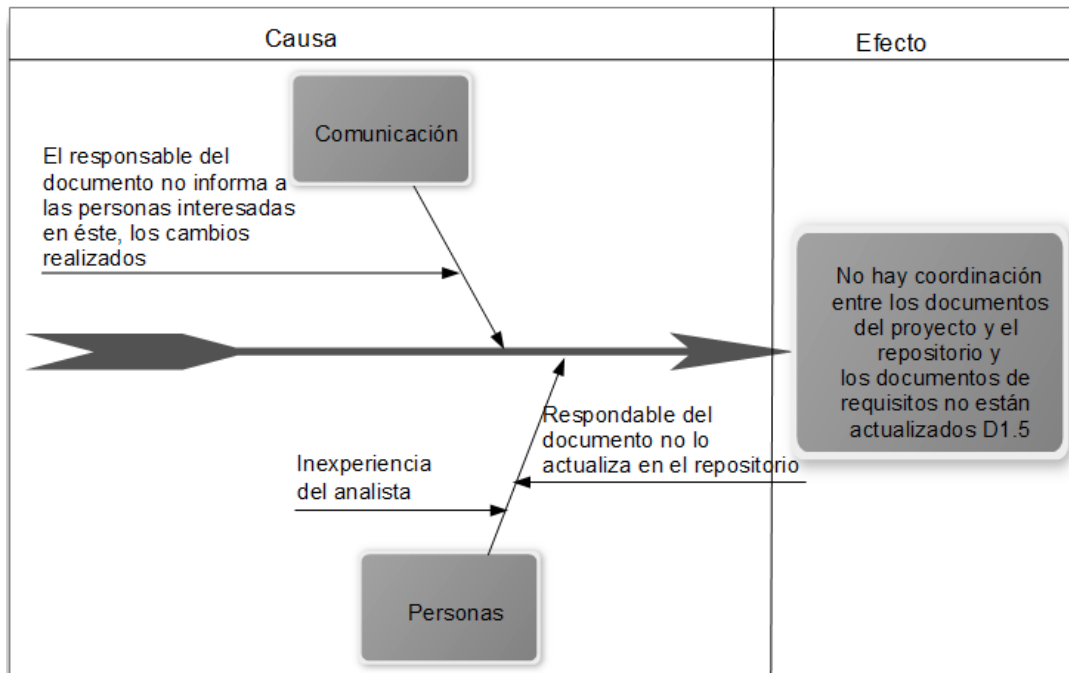


Figura 6. Diagrama de Ishikawa D1.5

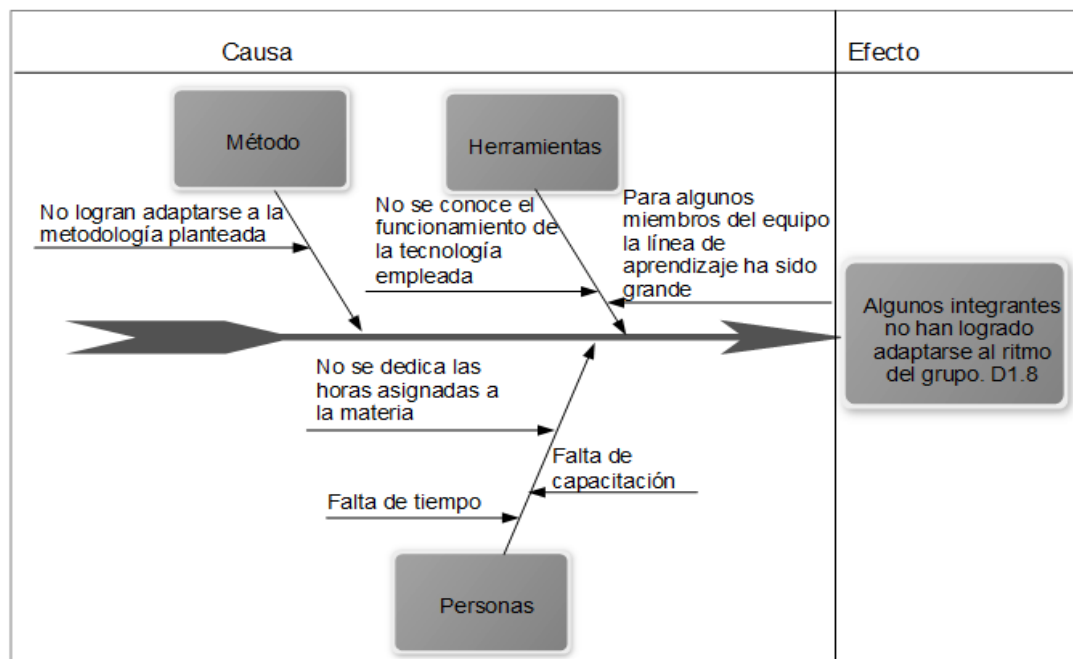


Figura 7. Diagrama de Ishikawa D1.8

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

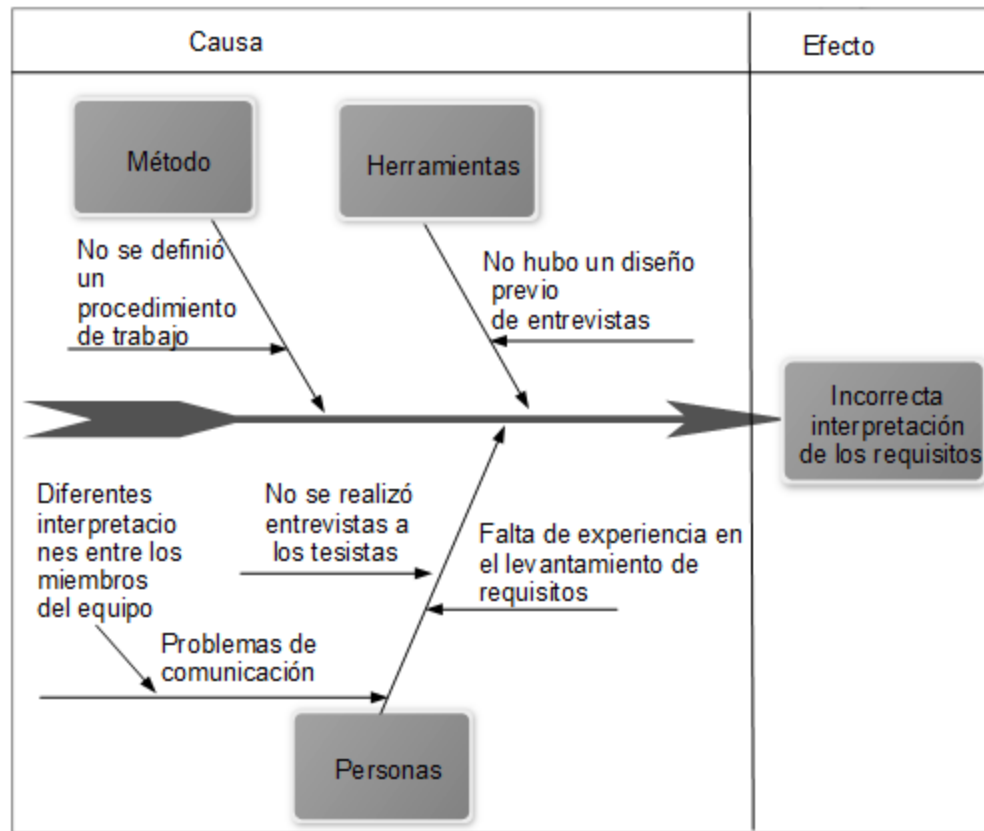


Figura 8. Diagrama de Ishikawa D1.7

2. Síntesis de las causas clasificadas en categorías

Defecto	Categorías				Otra categoría
	Herramientas	Entradas	Personas	Métodos	Comunicación
D1.4			X		X
D1.1			X		X
D1.2			X	X	
D1.3			X	X	
D1.5			X		X

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

D1.8	X		X	X	
D1.7	X		X	X	

❖ Causas fundamentales

Ficha del documento

Fecha	Revisión	Autor	Observaciones
17-09-2014		Kelly Yohanna Zuñiga	
Elaboró: Kelly Yohanna Zuñiga Fecha: 17-09-2014		Revisó: Kelly Yohanna Zuñiga Fecha: 17-09-2014	

Propósito

El propósito de este documento es mantener evidencias de las causas que generan mayor impacto en el desarrollo del proyecto.

El documento contiene:

1. Causas fundamentales

Defecto	Causa
D1.4	<ul style="list-style-type: none"> ❖ No se ha establecido un medio para acordar las fechas de las reuniones ❖ Falta de motivación
D1.1	<ul style="list-style-type: none"> ❖ No Se ha establecido fechas límites para las entregas ❖ Falta de responsabilidad
D1.2	<ul style="list-style-type: none"> ❖ No se ha definido de forma clara los mecanismos de

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

	comunicación
D1.3	<ul style="list-style-type: none"> ❖ El gerente del proyecto no ha establecido un mecanismo para asignar tareas ❖ No se reporta la finalización de las tareas
D1.5	<ul style="list-style-type: none"> ❖ El documento no es actualizado en el repositorio ❖ El responsable de un cambio no informa a los involucrados en el documento, sobre los cambios realizados.
D1.8	<ul style="list-style-type: none"> ❖ No se dedican las horas asignadas a las materia ❖ No se conoce el funcionamiento de la tecnología empleada
D1.7	<ul style="list-style-type: none"> ❖ Falta de experiencia en el levantamiento de requisitos ❖ Diferentes interpretaciones entre los miembros del equipo

2. Gráfico radial

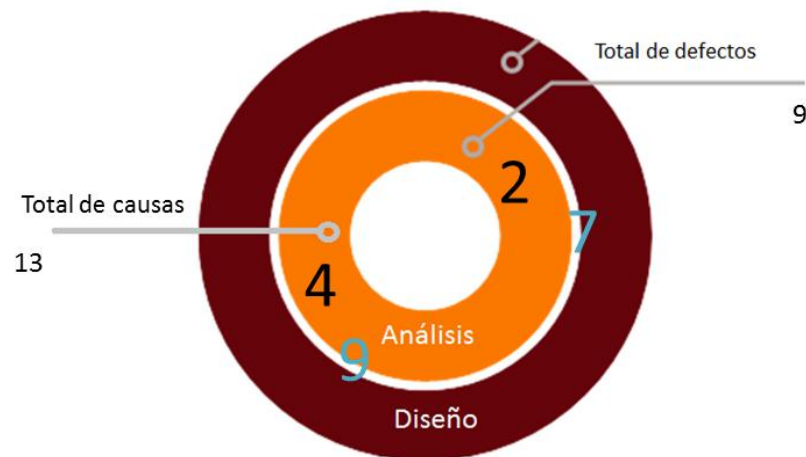


Figura 9. Diagrama radial fase 1

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

Según lo mostrado en la Figura 9, la disciplina análisis tiene directamente relacionado dos defectos (D1.5, D1.7), además de los siete defectos restantes que se encuentran tanto en análisis como en diseño. Esto se debe a que para este momento los defectos que se habían identificado estaban relacionados con la ejecución del proyecto, puesto que los responsables del desarrollo no habían empezado a programar los diferentes módulos y aun no se habían validado los modelos y diagramas realizados en el proyecto.

❖ Desarrollar recomendaciones

PLANTILLA DE RECOMENDACIONES				
PROCEDIMIENTO DE ANÁLISIS CAUSAL ENFOCADO EN PEQUEÑAS ORGANIZACIONES DE DESARROLLO SOFTWARE (PAC-DS)				
Organización	Grupo proyecto II			
Proyecto	Sistemas de gestión de actividades diarias-PAD			
Autor de las recomendaciones: Kelly Yohanna Zuñiga Silva				
Beneficios esperados al implementar las recomendaciones				
<ul style="list-style-type: none"> ❖ Mejorar la comunicación entre los miembros del equipo ❖ Que las tareas sean entregadas en las fechas establecidas ❖ Aumentar el conocimiento que tienen los desarrolladores en el framework ❖ Aclarar la funcionalidad requerida por el cliente 				
ID	Causa	Recomendaciones	Responsable	Fecha límite
R1.1	No se ha establecido un medio para acordar las reuniones	Establecer el medio por el cual se notifica a los integrantes del proyecto los cambios en las fechas de las reuniones o días adicionales en que se deben reunir.	Willian Fernando Pantoja	22-09-2014

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

R1.2	Falta de motivación	Durante las reuniones de análisis causal el gerente del proyecto debe dar a los miembros del equipo una breve charla con el propósito de motivarlos durante el proyecto	Willian Fernando Pantoja	Fechas establecidas para las reuniones de análisis causal
R1.3	No se ha establecido fechas límites para las entregas.	Realizar una plantilla donde se especifique por semana las tareas que se deben realizar, el responsable y la fecha límite de entrega. Ésta debe ser publicada semanalmente.	Willian Fernando Pantoja	22-09-2014
R1.4	Falta de responsabilidad	En caso de ser necesario se debe informar de esta situación al asesor del proyecto Libardo Pantoja	Willian Fernando Pantoja	Durante la ejecución del proyecto
R1.5	No se ha definido los mecanismos de comunicación	Definir la red social o medio de comunicación por la cual se va a informar sobre las fechas de las reuniones, cambios realizados y tareas terminadas.	Willian Fernando Pantoja	22-09-2014
R1.6	No se reporta la finalización de tareas	Cada miembro del equipo debe colocar en el medio de comunicación establecido, las tareas que ha finalizado, esto con el fin de que se le pueda asignar nuevas tareas.	Líder del grupo (líder de calidad, de desarrollo o gerente del proyecto)	Durante la ejecución del proyecto
R1.7	El documento no es actualizado en el repositorio	Cada miembro del equipo, al realizar un cambio en un documento también debe de actualizarlo en el repositorio (Google Drive)	Responsable del documento	Durante la ejecución del proyecto
R1.8	No se informa a los involucrados sobre los	Cada miembro del equipo, al realizar un cambio en un documento también debe notificarlo mediante el medio de	Responsable del documento	Durante la ejecución del proyecto

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

	cambios realizados	comunicación establecido		
R1.9	No se conoce la tecnología empleada	El gerente del proyecto debe programar unas horas para capacitar a los desarrollares en el framework Yii	Willian Fernando Pantoja	2-10-204
	Diferentes interpretaciones entre los miembros del equipo	El analista debe reunirse con el cliente a fin de aclarar ambigüedades sobre los requisitos, mediante el uso de prototipos de papel.	Andrea Zambrano	22-09-2014
Elaboró: Kelly Yohanna Zuñiga Fecha: 18-09-2014		Revisó: Kelly Yohanna Zuñiga Fecha: 18-09-2014		

❖ Seguimiento a las recomendaciones

PLANTILLA DE SEGUIMIENTO DE RECOMENDACIONES			
PROCEDIMIENTO DE ANÁLISIS CAUSAL ENFOCADO EN PEQUEÑAS ORGANIZACIONES DE DESARROLLO SOFTWARE (PAC-DS)			
Organización	Grupo proyecto II		
Proyecto	Sistemas de gestión de actividades diarias-PAD		
ID	Fecha	Verificación del cumplimiento	Fecha de cierre
R1.1	23-09-2014	El gerente del proyecto estableció e informó a todos los miembros del equipo que el medio de comunicación sería la red social Facebook.	23-09-2014
R1.3	23-09-2014	El gerente del proyecto elabora una plantilla donde se define tareas, responsable y fecha límite de entrega.	23-09-2014

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

R1.5	23-09-2014	El gerente del proyecto estableció como mecanismo de comunicación la red social Facebook y como repositorio de documentación Google Drive.	23-09-2014
R1.6	25-09-2014	A pesar de la recomendación algunos miembros no informan las sobre las tareas finalizadas.	18-17-2014
R1.7	25-09-2014	A pesar de la recomendación algunos miembros no actualizan los documentos en el repositorio	Inconveniente presentado hasta el final del proyecto
R1.8	25-09-2014	A pesar de la recomendación en algunas ocasiones no se informa sobre los cambios realizados	17-10-2014
R1.9	2-10-2014	No se ha cumplido con esta recomendación	18-10-2014
R1.10	23-09-2014	Se aclaró dudas sobre algunas funcionalidades requeridas	23-09-2014
Elaboró: Kelly Yohanna Zuñiga Fecha: 22-09-2014		Revisó: Kelly Yohanna Zuñiga Fecha: 22-09-2014	

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

SEGUNDA FASE

1. Actividad detección de defectos

❖ Recolección de datos

PLANTILLA DE RECOLECCIÓN DE DEFECTOS		
PROCEDIMIENTO DE ANÁLISIS CAUSAL ENFOCADO EN PEQUEÑAS ORGANIZACIONES DE DESARROLLO SOFTWARE (PAC-DS)		
Organización	Grupo proyecto II	
Proyecto	Sistemas de gestión de actividades diarias-PAD	
Fecha: 06 se octubre del 2014		
ID	Defecto	Cantidad de personas que lo reportaron
D2.1	Falta de conocimiento del framework	1
D2.2	Retrasos en la entrega del primer modulo	2
D2.3	No se ha implementado la plantilla de la aplicación(retrasos)	1
D2.4	Los desarrolladores no han seguido la secuencia establecida en los prototipos	1
D2.5	Los miembros del equipo no revisan la documentación del proyecto	1
D2.6	El programa tiene información irrelevante que generaba ruido y no se está empleando	2
D2.7	Se asiste a la reuniones, pero no se trabaja en lo relacionado con Proyecto II	3

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

D2.8	Al ingresar datos inválidos para crear una nueva categoría, no se muestra un mensaje indicando que el nombre de la categoría es obligatorio	2
D2.9	El sistema no cuenta con la funcionalidad de editar categoría	2
D2.10	El sistema no cuenta con la funcionalidad de eliminar categoría	2
D2.11	El sistema permite ingresar datos inválidos para crear una actividad	2
D2.12	El sistema no permite editar una actividad	2
D2.13	El sistema no permite eliminar una actividad	2
D2.14	Crear tarea no cumple con la especificación de los prototipos	2
D2.15	Modificar el idioma (de inglés a español) específicamente el del calendario	2
D2.16	Al crear una tarea se debería mostrar estándar de horario (ejemplo: dd/mes/año)	2
D2.17	El sistema no confirma si la tarea fue o no guardada	2
D2.18	A la hora de crear una tarea, el botón cancelar no tiene funcionalidad	2
D2.19	No se indica que campos son obligatorios para crear una tarea	2
D2.20	No se puede editar una tarea	2
D2.21	El sistema no cuenta con la funcionalidad de eliminar tarea	2
DIAGRMA DE AFINIDAD		

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

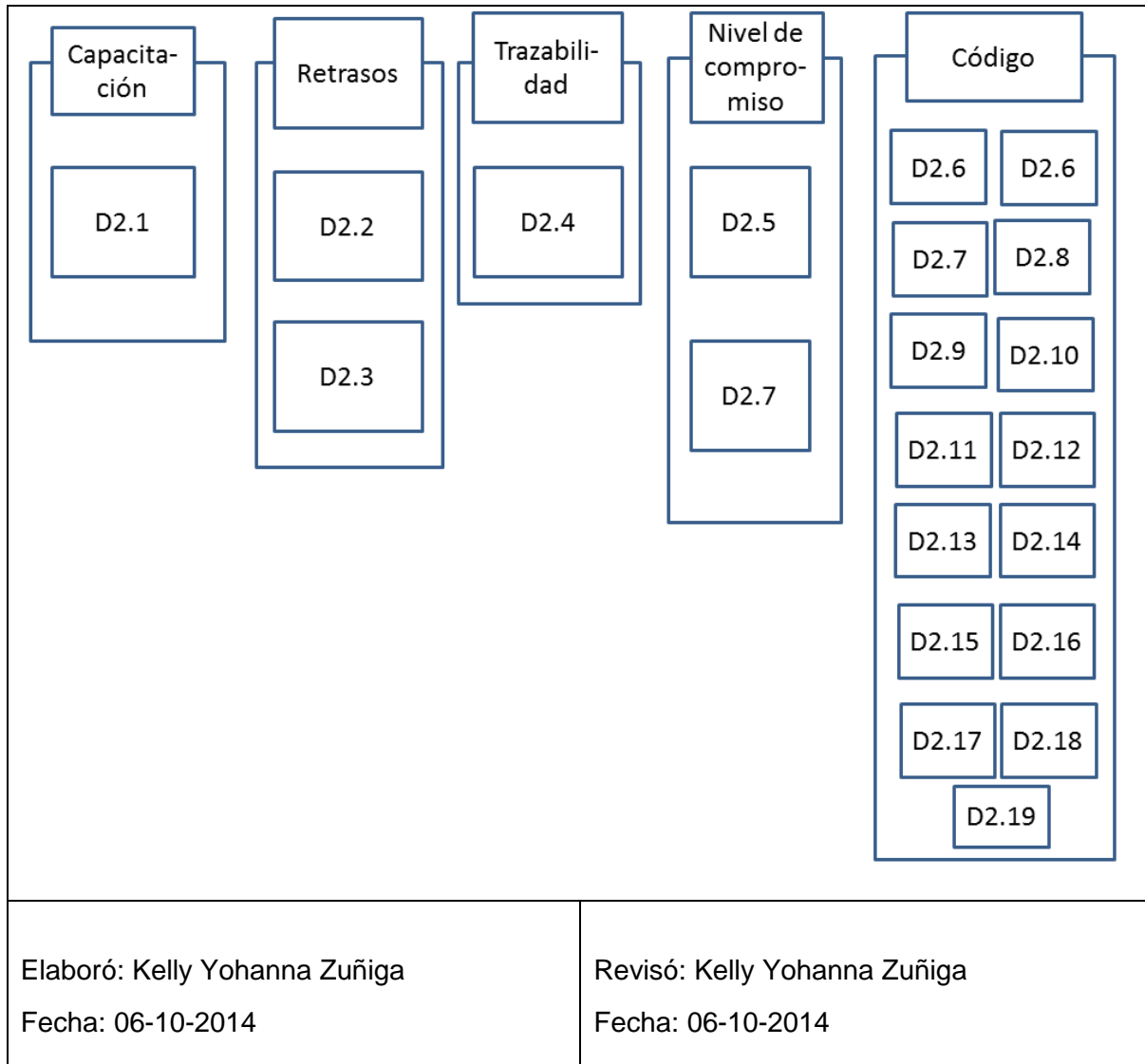


Tabla 30. Plantilla de recolección de defectos, fase 2

❖ Determinar muestra de los defectos

Una vez se tienen los defectos que han surgido durante la ejecución del proyecto es necesario realizar el diagrama de Pareto para determinar la muestra de los defectos. Este fue realizado de acuerdo a la Tabla 30.

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

Ficha del documento

Fecha	Revisión	Autor	Observaciones
06-10-2014		Kelly Yohanna Zuñiga	
Elaboró: Kelly Yohanna Zuñiga Fecha: 06-10-2014		Revisó: Kelly Yohanna Zuñiga Fecha: 06-10-2014	

Propósito

El propósito de este documento es mantener evidencias de los defectos que generan mayor impacto en el desarrollo del proyecto.

El documento contiene:

1. Diagrama de Pareto o análisis modal de fallo y efecto

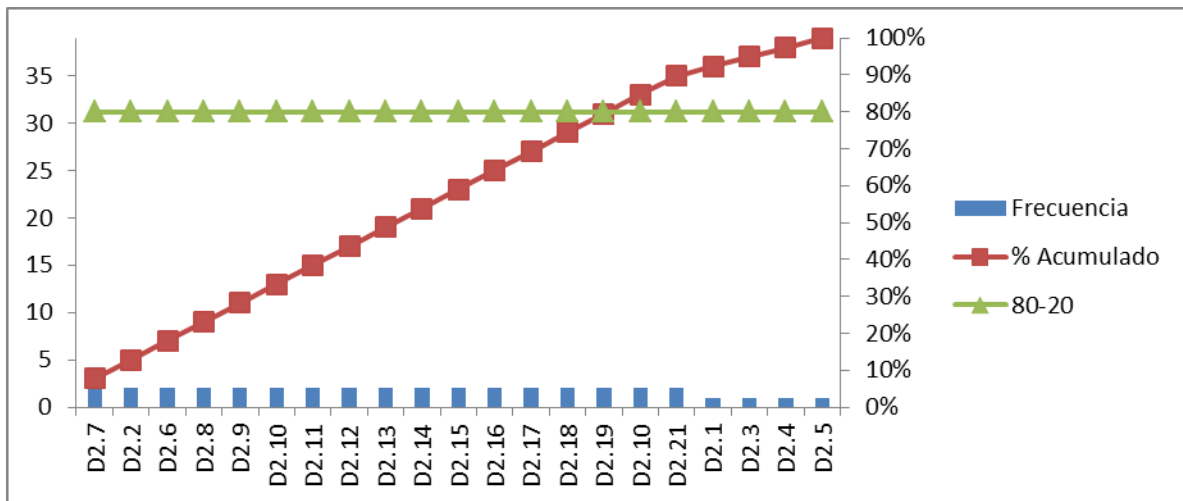


Figura 10. Diagrama de Pareto para los defectos detectados en la fase 2

Según lo mostrado en el diagrama de Pareto resultante (Figura 10), el 80% de estos defectos se deben a un 20% de causas que deben ser eliminadas. Para este propósito la línea de color

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

rojo (% acumulado) debajo de la línea verde, muestra los defectos que deben ser abordados para su respectivo análisis.

2. Defectos de mayor impacto

Identificador	Defecto
D2.7	Se asiste a la reuniones, pero no se trabaja en lo relacionado con Proyecto II
D2.2	Retrasos en la entrega del primer módulo
D2.6	El programa tiene información irrelevante que generaba ruido y no se está empleando
D2.8	Al ingresar datos inválidos para crear una nueva categoría, no se muestra un mensaje indicando que el nombre de la categoría es obligatorio
D2.10	El sistema no cuenta con la funcionalidad de eliminar categoría
D2.11	El sistema permite ingresar datos inválidos para crear una actividad
D2.12	El sistema no permite editar una actividad
D2.13	El sistema no permite eliminar una actividad
D2.14	Crear tarea no cumple con la especificación de los prototipos
D2.15	Modificar el idioma (de inglés a español) específicamente en el calendario
D2.16	Al crear una tarea se debería mostrar estándar de horario (ejemplo: dd/mes/año)
D2.17	El sistema no confirma si la tarea fue o no guardada
D2.18	A la hora de crear una tarea, el botón cancelar no tiene funcionalidad

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

❖ Clasificar los defectos

Se procede a clasificar los defectos relacionados directamente con el producto software.

Defecto: Al ingresar datos inválidos para crear una nueva categoría, no se muestra un mensaje indicando que el nombre de la categoría es obligatorio D2.8			
Fecha: 07-10-2014		Fecha: 10-11-2014	
Etapas de apertura		Etapas de cierre	
Actividad	Pruebas funcionales	Origen	Código
Disparador	Cobertura	Tipo de defecto	Comprobación
Impacto para el cliente	Capacidad	Calificador	Incorrecta
		Fuente	Desarrollado In-House
		Edad	Nueva

Defecto: El sistema no cuenta con la funcionalidad de eliminar categoría D2.10			
Fecha: 07-10-2014		Fecha: 10-11-2014	
Etapas de apertura		Etapas de cierre	
Actividad	Pruebas funcionales	Origen	Código
Disparador	Cobertura	Tipo de defecto	Inicialización
Impacto para el cliente	Capacidad	Calificador	Que falta (Missing)
		Fuente	Desarrollado In-House
		Edad	Nueva

Defecto: El sistema permite ingresar datos inválidos para crear una actividad D2.11
--

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

Fecha: 07-10-2014		Fecha: 10-11-2014	
Etapa de apertura		Etapa de cierre	
Actividad	Pruebas funcionales	Origen	Código
Disparador	Cobertura	Tipo de defecto	Comprobación
Impacto para el cliente	Capacidad	Calificador	Incorrecta
		Fuente	Desarrollado In-House
		Edad	Nueva

Defecto: El sistema no permite editar una actividad D2.12			
Fecha: 07-10-2014		Fecha: 10-11-2014	
Etapa de apertura		Etapa de cierre	
Actividad	Pruebas funcionales	Origen	Código
Disparador	Cobertura	Tipo de defecto	Inicialización
Impacto para el cliente	Capacidad	Calificador	Que falta (Missing)
		Fuente	Desarrollado In-House
		Edad	Nueva

Defecto: El sistema no permite eliminar una actividad D2.13			
Fecha: 07-10-2014		Fecha: 10-11-2014	
Etapa de apertura		Etapa de cierre	
Actividad	Pruebas funcionales	Origen	Código
Disparador	Cobertura	Tipo de defecto	Inicialización
Impacto para el	Capacidad	Calificador	Que falta (Missing)

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

cliente			
		Fuente	Desarrollado In-House
		Edad	Nueva

Defecto: Crear tarea no cumple con la especificación de los prototipos D2.14			
Fecha: 07-10-2014		Fecha: 03-11-2014	
Etapa de apertura		Etapa de cierre	
Actividad	Revisar diseño	Origen	Código
Disparador	Cobertura	Tipo de defecto	Función
Impacto para el cliente	Capacidad	Calificador	Que falta (Missing)
		Fuente	Desarrollado In-House
		Edad	Nueva

Defecto: Modificar el idioma (de inglés a español) específicamente en el calendario D2.15			
Fecha: 07-10-2014		Fecha: 24-11-2014	
Etapa de apertura		Etapa de cierre	
Actividad	Revisar GUI	Origen	Soporte de lenguaje nacional
Disparador	Cobertura	Tipo de defecto	Traducción
Impacto para el cliente	Capacidad	Calificador	Que falta (Missing)
		Fuente	Desarrollado In-House
		Edad	Nueva

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

Defecto: Al crear una tarea se debería mostrar estándar de horario (ejemplo: dd/mes/año) D2.16			
Fecha: 07-10-2014		Fecha: 24-11-2014	
Etapa de apertura		Etapa de cierre	
Actividad	Revisar GUI	Origen	Información de desarrollo
Disparador	Conformidad de diseño	Tipo de defecto	Editorial
Impacto para el cliente	Capacidad	Calificador	Que falta (Missing)
		Fuente	Desarrollado In-House
		Edad	Nueva

Defecto: El sistema no confirma si la tarea fue o no guardada D2.17			
Fecha: 07-10-2014		Fecha: 10-11-2014	
Etapa de apertura		Etapa de cierre	
Actividad	Pruebas funcionales	Origen	Código
Disparador	Cobertura	Tipo de defecto	Inicialización
Impacto para el cliente	Capacidad	Calificador	Que falta (Missing)
		Fuente	Desarrollado In-House
		Edad	Nueva

Defecto: A la hora de crear una tarea, el botón cancelar no tiene funcionalidad D1.18			
Fecha: 07-10-2014		Fecha: 10-11-2014	
Etapa de apertura		Etapa de cierre	
Actividad	Pruebas funcionales	Origen	Código

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

Disparador	Cobertura	Tipo de defecto	Inicialización
Impacto para el cliente	Capacidad	Calificador	Que falta (Missing)
		Fuente	Desarrollado In-House
		Edad	Nueva

2. Actividad Detección de causas fundamentales

❖ Identificar causas

Ficha del documento

Fecha	Revisión	Autor	Observaciones
07-10-2014		Kelly Yohanna Zuñiga	
Elaboró: Kelly Yohanna Zuñiga Fecha: 07-10-2014		Revisó: Kelly Yohanna Zuñiga Fecha: 07-10-2014	

Propósito

El propósito de este documento es tener información histórica que permita comparar situaciones perjudiciales presentadas previamente y evitar su aparición en proyectos futuros. Asimismo este reporte especifica el responsable de realizar los diagramas correspondientes.

El documento contiene:

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

1. Diagramas

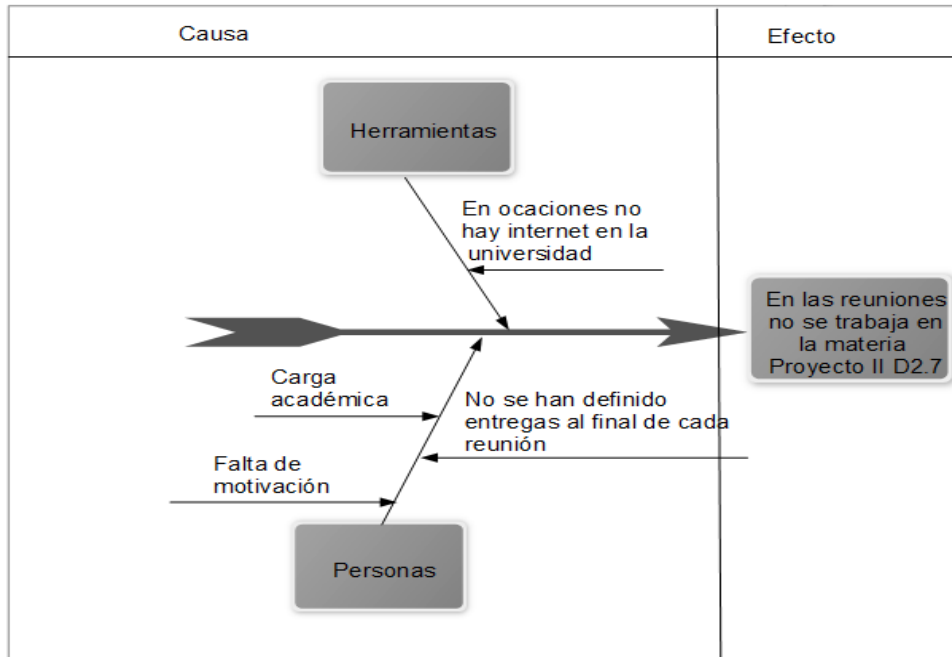


Figura 11. Diagrama de Ishikawa D2.7

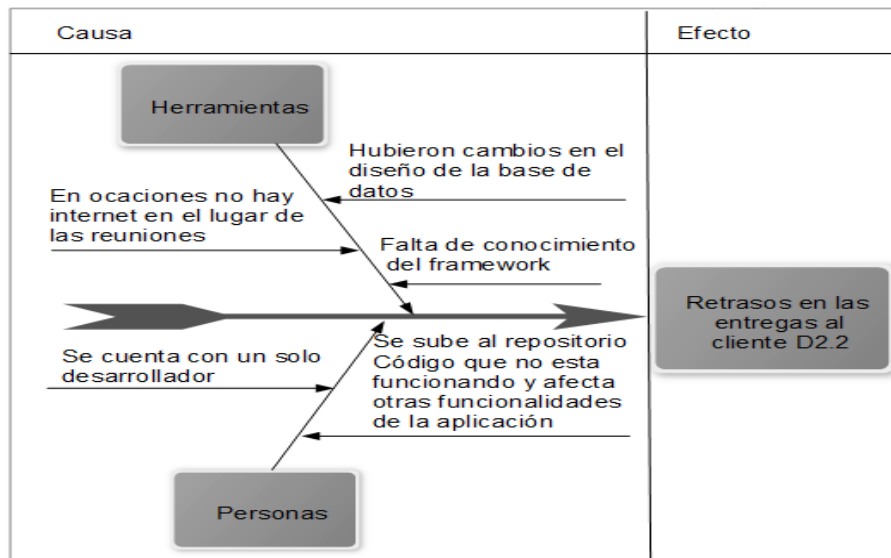


Figura 12. Diagrama de Ishikawa D2.2

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

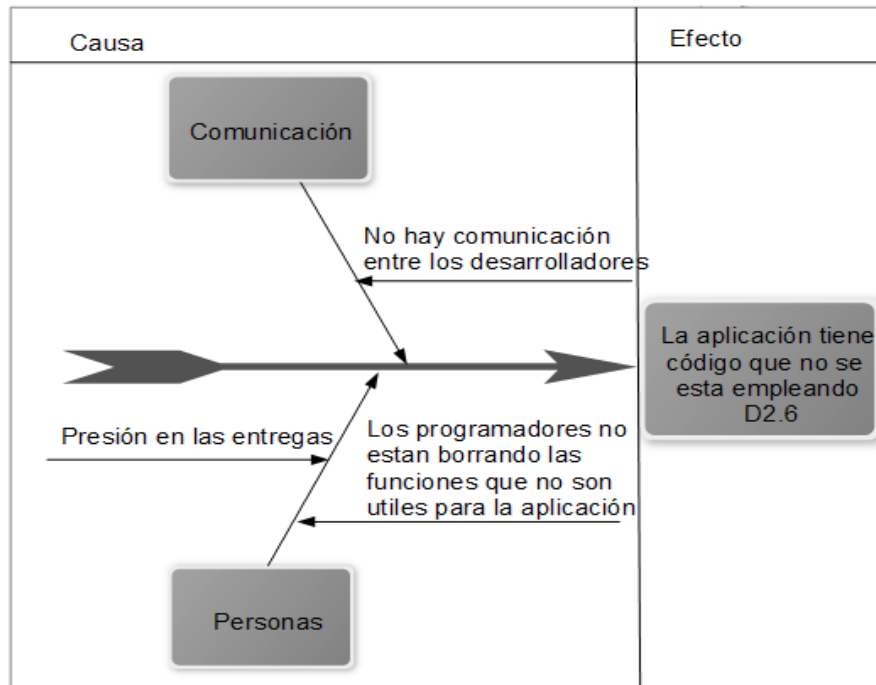


Figura 13. Diagrama de Ishikawa D2.6

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

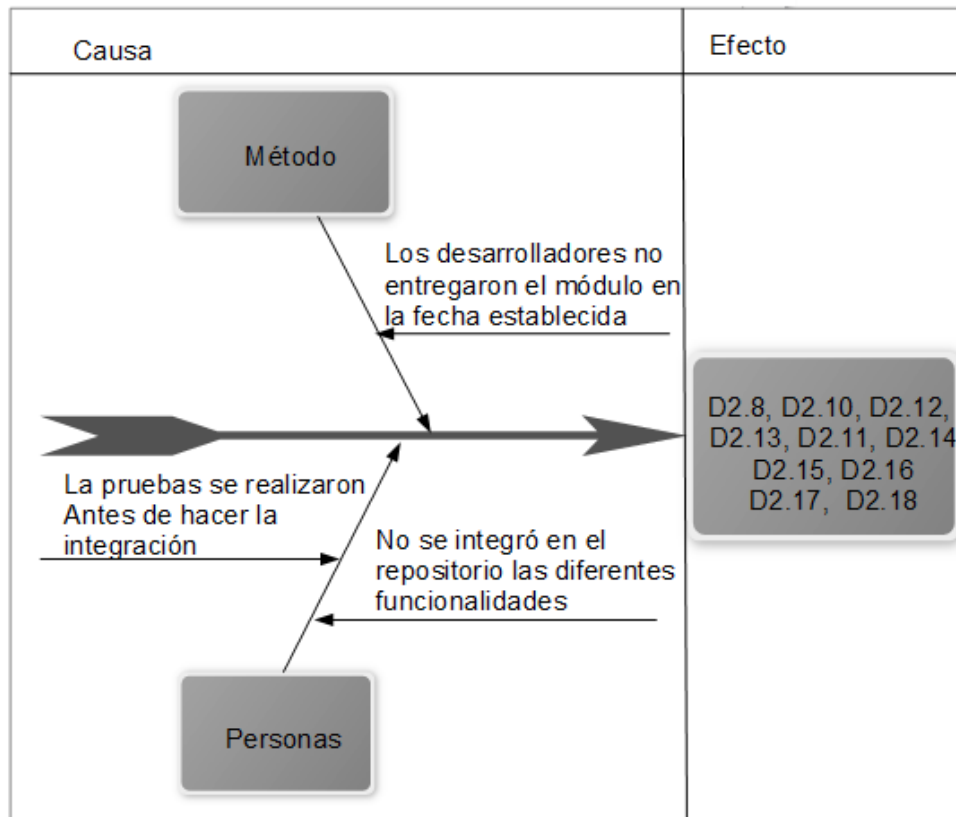


Figura 14. Defectos relacionados con el producto software

2. Síntesis de las causas clasificadas en categorías

Defecto	Categorías				Otra categoría
	Herramientas	Entradas	Personas	Métodos	Comunicación
D2.7	X		X		
D2.2	X		X		
D2.6	X				X
D2.8, D2.10, D2.11, D2.12, D2.13, D2.14, D2.15, D2.16, D2.17, D2.18			X	X	

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

❖ Causas fundamentales

Ficha del documento

Fecha	Revisión	Autor	Observaciones
07-10-2014		Kelly Yohanna Zuñiga	
Elaboró: Kelly Yohanna Zuñiga Fecha: 07-10-2014		Revisó: Kelly Yohanna Zuñiga Fecha: 07-10-2014	

Propósito

El propósito de este documento es mantener evidencias de las causas que generan mayor impacto en el desarrollo del proyecto.

El documento contiene:

1. Causas fundamentales

Defecto	Causa
D2.7	<ul style="list-style-type: none"> ❖ No se ha definido entregas al final de cada reunión ❖ En ocasiones no hay internet en la universidad
D2.2	<ul style="list-style-type: none"> ❖ Se cuenta con un solo desarrollador, además del desarrollador Front End ❖ Hubo cambios en el diseño de la base de datos
D2.6	<ul style="list-style-type: none"> ❖ Los programadores no están borrando las funciones que no son útiles para la aplicación
D2.8, D2.10, D2.11, D2.12, D2.13, D2.14, D2.15, D2.16, D2.17, D2.18	<ul style="list-style-type: none"> ❖ Las pruebas se realizaron antes de hacer la integración

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

2. Gráfico radial

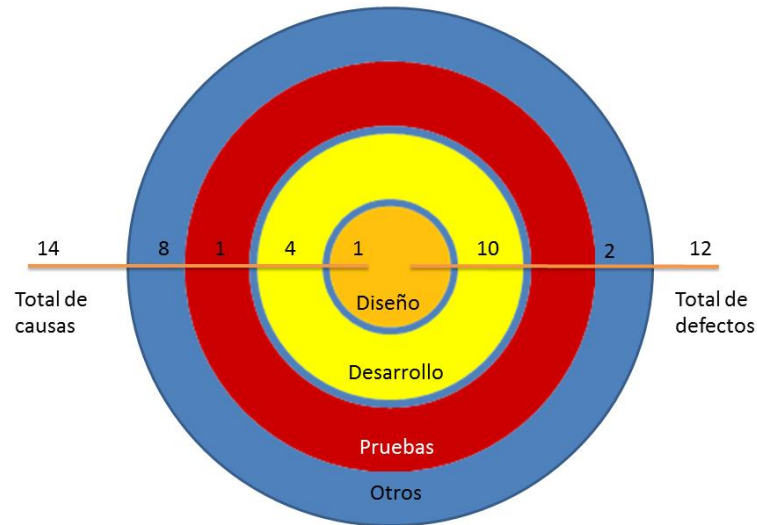


Figura 15. Diagrama radial fase 2

Según lo mostrado en la **¡Error! No se encuentra el origen de la referencia.** se presentó una mayor cantidad de defectos en la disciplina de desarrollo (D1.6, D1.8, D1.10, D1.11, D1.12, D1.13, D1.14, D1.15, D1.16, D1.17, D1.18), lo cual indica que se debe colocar énfasis en esta disciplina.

❖ Desarrollar recomendaciones

PLANTILLA DE RECOMENDACIONES	
PROCEDIMIENTO DE ANÁLISIS CAUSAL ENFOCADO EN PEQUEÑAS ORGANIZACIONES DE DESARROLLO SOFTWARE (PAC-DS)	
Organización	Grupo proyecto II
Proyecto	Sistemas de gestión de actividades diarias-PAD
Autor de las recomendaciones: Kelly Yohanna Zuñiga Silva	
Beneficios esperados al implementar las recomendaciones	

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

ID	Causa	Recomendaciones	Responsable	Fecha límite
R2.1	No se definen entregas al final de cada reunión	Se recomienda al gerente del proyecto definir que tareas deben ser entregadas al final de las reuniones que se realizan durante la semana	William Pantoja	13-10-2014
R2.2	No hay internet en el lugar de reunión	Se recomienda que los miembros del equipo puedan trabajar desde cualquier lugar, siempre y cuando entreguen las tareas asignadas.	Responsable de la tarea	Durante el desarrollo del proyecto
R2.3	Se cuenta con un solo desarrollador	Algunos miembros del equipo tienen que apoyar a los miembros del equipo de desarrollo	William Pantoja	13-10-2014
R2.4	Los programadores no están borrando las funciones que no son útiles	Asignar a un miembro del equipo la tarea para limpiar el código de la aplicación	William Pantoja	15-10-2014
R2.5	Causas relacionadas con los defectos presentados en el producto software	<p>Cada desarrollador debe asegurarse de que el código subido al repositorio no ha afectado otras funcionalidades</p> <p>Las pruebas se deben hacer antes de realizar la integración, en caso de presentarse retrasos por parte de los desarrolladores es preferible aplazar las pruebas.</p> <p>El desarrollador Fron End debe reunirse como mínimo una vez</p>	Desarrolladores	Durante el desarrollo del proyecto

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

		<p>por semana con los otros desarrolladores a fin de que coordine actividades.</p> <p>Nombrar un responsable para implementar los mensajes de confirmación que requiere la aplicación</p>	<p>Andrea Zambrano</p>	<p>14-10-2014</p>
<p>Elaboró: Kelly Yohanna Zuñiga Fecha: 09-10-2014</p>		<p>Revisó: Kelly Yohanna Zuñiga Fecha: 09-10-2014</p>		

❖ Seguimiento a las recomendaciones

PLANTILLA DE SEGUIMIENTO DE RECOMENDACIONES			
PROCEDIMIENTO DE ANÁLISIS CAUSAL ENFOCADO EN PEQUEÑAS ORGANIZACIONES DE DESARROLLO SOFTWARE (PAC-DS)			
Organización	Grupo proyecto II		
Proyecto	Sistemas de gestión de actividades diarias-PAD		
ID	Fecha	Verificación del cumplimiento	Fecha de cierre
R2.1	13-10-2014	El gerente del proyecto definió las tareas que debían ser cumplidas en cada reunión	13-10-2014
R2.2	Durante la ejecución del proyecto	Los miembros del equipo trabajan desde otros lugares, sin embargo persisten los problemas de las entregas en las fecha indicadas	En la fase final se cumplió con la recomendación, a partir del 04-11-2014
R2.3	13-10-2014	El gerente del proyecto decidió apoyar al equipo de desarrollo, lo	13-10-2014

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

		cual mitigo los retrasos llevados hasta la fecha	
R2.4	15-10-2014	Se eliminó código que no era útil para la aplicación	15-10-2014
R2.5	Durante la ejecución del proyecto	Las recomendaciones fueron aplicadas, hubo mayor comunicación entre los desarrolladores el desarrollador Front End, además los mensajes fueron implementados	Durante la ejecución del 14-10-2014
Elaboró: _____ Fecha: _____		Revisó: _____ Fecha: _____	

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

REFERENCIAS

- [1] K. Wong, "Using an Ishikawa diagram as a tool to assist memory and retrieval of relevant medical cases from the medical literature," *J. Med. Case Rep.*, vol. 5, no. 1, pp. 1–3, 2011.
- [2] A. Jayswal, X. Li, A. Zanwar, H. H. Lou, and Y. Huang, "A sustainability root cause analysis methodology and its application," *Comput. Chem. Eng.*, vol. 35, no. 12, pp. 2786–2798, Dec. 2011.
- [3] J. E. J. Gibson, D. U. S. Power, and T. Group, "Cause and effect analysis: 'Power tool' for total quality," *Electr. Electron. Insul. Conf. Electr. Manuf. Coil Wind. Conf.*, p. 751,753, 1993.
- [4] M. E. Cournoyer, S. Trujillo, S. Schreiber, M. T. Saba, and M. C. Peabody, "Causal analysis of a glovebox glove breach," *J. Chem. Heal. Saf.*, vol. 20, no. 2, pp. 25–33, Mar. 2013.
- [5] P. Henry, "The Analysis of Defect Root Causes," in *The Testing Network SE - 7*, Springer Berlin Heidelberg, 2008, pp. 235–296.
- [6] I. C. Mario Piattini, Félix Garcia, *Calidad de sistemas informáticos*. 2007, p. 388.
- [7] H. A. Shari, N. Khalid, N. S. Ashaari, and H. M. Judi, "Statistical Process Control in Plastic Packaging Manufacturing : A Case Study," *Electr. Eng. Informatics, 2009. ICEEI '09. Int. Conf.*, vol. 01, no. August, pp. 199–203, 2009.
- [8] W. Semke, K. Lemler, and M. Thapa, "An Experimental Modal Channel Reduction Procedure Using a Pareto Chart," in *Topics in Modal Analysis II, Volume 8 SE - 10*, R. Allemang, Ed. Springer International Publishing, 2014, pp. 101–110.
- [9] E. Cano, J. Moguerza, and A. Redchuk, "Pareto Analysis with R," in *Six Sigma with R SE - 6*, vol. 36, Springer New York, 2012, pp. 91–100.
- [10] M. J. Eppler, "A comparison between concept maps, mind maps, conceptual diagrams, and visual metaphors as complementary tools for knowledge construction and sharing," *Inf. Vis.*, vol. 5, no. 3, pp. 202–210, Jun. 2006.
- [11] L. Buglione, "Strengthening CMMI Maturity Levels with a Quantitative Approach to Root-Cause Analysis," *Proc. 5th Softw. Meas. Eur. Forum (SMEF 2008), Ital. Milan*, pp. 67–82, 2008.

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

- [12] M. Crowe and L. Sheppard, "Mind mapping research methods," *Qual. Quant.*, vol. 46, no. 5, pp. 1493–1504, 2012.
- [13] M. Davies, "Concept mapping, mind mapping and argument mapping: what are the differences and do they matter?," *High. Educ.*, vol. 62, no. 3, pp. 279–301, 2011.
- [14] P. Farrand, F. Hussain, and E. Hennessy, "The efficacy of the 'mind map' study technique.," *Med. Educ.*, vol. 36, no. 5, pp. 426–31, May 2002.
- [15] Tony Buzan, *Cómo crear Mapas Mentales*, Thorsons,. 2004.
- [16] M. Quaddus and A. Intrapairot, "Management policies and the diffusion of data warehouse: a case study using system dynamics-based decision support system," *Decis. Support Syst.*, vol. 31, no. 2, pp. 223–240, Jun. 2001.
- [17] J. D. Sterman, "System Dynamics Modeling: TOOLS FOR LEARNING IN A COMPLEX WORLD.," *Calif. Manage. Rev.*, vol. 43, no. 4, p. 8, 2001.
- [18] A. Mansourian and E. Abdolmajidi, "Investigating the system dynamics technique for the modeling and simulation of the development of spatial data infrastructures.," *Int. J. Geogr. Inf. Sci.*, vol. 25, no. 12, pp. 2001–2023, Dec. 2011.
- [19] S. Arreche and S. Matalonga, "Tools for defect causal analysis," *Inf. Syst. Technol. (CISTI), 2012 7th Iber. Conf.*, p. 1,7, 2012.
- [20] J. Tejada and S. Ferreira, "Applying Systems Thinking to Analyze Wind Energy Sustainability," *Procedia Comput. Sci.*, vol. 28, no. 0, pp. 213–220, 2014.
- [21] J. W. Forrester, "System dynamics, systems thinking, and soft OR," *Syst. Dyn. Rev.*, vol. 10, no. January, pp. 245–256, 1994.
- [22] V. P. Gurupur, U. Sakoglu, G. P. Jain, and U. J. Tanik, "Semantic requirements sharing approach to develop software systems using concept maps and information entropy: A Personal Health Information System example," *Adv. Eng. Softw.*, vol. 70, no. 0, pp. 25–35, Apr. 2014.
- [23] J. Village, F. A. Salustri, and W. P. Neumann, "Cognitive mapping: Revealing the links between human factors and strategic goals in organizations," *Int. J. Ind. Ergon.*, vol. 43, no. 4, pp. 304–313, Jul. 2013.

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

- [24] K. Siau and X. Tan, "Improving the quality of conceptual modeling using cognitive mapping techniques," *Data Knowl. Eng.*, vol. 55, no. 3, pp. 343–365, Dec. 2005.
- [25] K. Siau, "Use of cognitive mapping techniques in information systems," *Inf. Manag.*, vol. 19, no. 3, pp. 18–21, 2006.
- [26] U. Özesmi and S. L. Özesmi, "Ecological models based on people's knowledge: a multi-step fuzzy cognitive mapping approach," *Ecol. Modell.*, vol. 176, no. 1–2, pp. 43–64, Aug. 2004.
- [27] R. Marshall, "Guest Editorial: Cognitive mapping of strategy in marketing," *J. Bus. Res.*, vol. 66, no. 9, pp. 1541–1543, Sep. 2013.
- [28] F. van Winsen, Y. de Mey, L. Lauwers, S. Van Passel, M. Vancauteran, and E. Wauters, "Cognitive mapping: A method to elucidate and present farmers' risk perception," *Agric. Syst.*, vol. 122, no. 0, pp. 42–52, Nov. 2013.
- [29] M. J. H. van Bon-Martens, L. a. M. van de Goor, J. C. Holsappel, T. J. M. Kuunders, M. a. M. Jacobs-van der Bruggen, J. H. M. te Brake, and J. a. M. van Oers, "Concept mapping as a promising method to bring practice into science," *Public Health*, Jun. 2014.
- [30] J. L. Gerdeman, K. Lux, and J. Jacko, "Using concept mapping to build clinical judgment skills," *Nurse Educ. Pract.*, vol. 13, no. 1, pp. 11–17, Jan. 2013.
- [31] M. A. Ruiz, A. Pardo, and S. Martín, "Sección Monográfica," *Papeles del psicólogo*, vol. 31, no. 1, pp. 34–45, 2010.
- [32] B. Xiong, M. Skitmore, B. Xia, M. A. Masrom, K. Ye, and A. Bridge, "Examining the influence of participant performance factors on contractor satisfaction: A structural equation model," *Int. J. Proj. Manag.*, vol. 32, no. 3, pp. 482–491, Apr. 2014.
- [33] I.-Y. Hsu, T.-S. Su, C.-S. Kao, Y.-L. Shu, P.-R. Lin, and J.-M. Tseng, "Analysis of business safety performance by structural equation models," *Saf. Sci.*, vol. 50, no. 1, pp. 1–11, Jan. 2012.
- [34] M. Cupani, "Análisis de Ecuaciones Estructurales: conceptos, etapas de desarrollo y un ejemplo de aplicación," *Rev. TESIS Fac. Psicol.*, vol. 2, no. 1, pp. 186–199, 2012.
- [35] M. A. G. Veiga, "Análisis causal con ecuaciones estructurales de la satisfacción ciudadana con los servicios municipales." pp. 1–116, 2011.

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

- [36] M. J. Wainwright and M. I. Jordan, "Graphical Models, Exponential Families, and Variational Inference," *Found. Trends® Mach. Learn.*, vol. 1, no. 1–2, pp. 1–305, 2007.
- [37] A. L. Bento and R. Bento, "The Use of Causal analysis techniques in information systems research: a methodological note," *J. Inf. Technol. Manag.*, vol. XV, no. 3, pp. 44–51, 2004.
- [38] C. Henderson, "Figure 1: Radial analysis chart," *ACM SIGSOFT Softw. Eng. Notes*, vol. 33, no. 4, p. 1, 2008.
- [39] C. Henderson, "Managing software defects: defect analysis and traceability," *ACM SIGSOFT Softw. Eng. Notes*, vol. 33, no. 4, p. 1, Jul. 2008.
- [40] T. T. Moores, "Developing a software size model for rule-based systems: a case study," *Expert Syst. Appl.*, vol. 21, no. 4, pp. 229–237, Nov. 2001.
- [41] E. Reformat, M. Igside, "Isolation of Software Defects : Extracting Knowledge with Confidence," *Inf. Reuse Integr. Conf, 2005. IRI -2005 IEEE Int. Conf.*, pp. 114–119, 2005.
- [42] P. D. Chatzoglou and L. A. Macaulay, "A Rule-Based Approach to Developing Software Development Prediction Models," *Autom. Softw. Eng.*, vol. 5, no. 2, pp. 211–243, 1998.
- [43] J. Chai, J. N. K. Liu, and Z. Xu, "A rule-based group decision model for warehouse evaluation under interval-valued Intuitionistic fuzzy environments," *Expert Syst. Appl.*, vol. 40, no. 6, pp. 1959–1970, May 2013.
- [44] B. Chaib-Draa and J. Desharnais, "A relational model of cognitive maps," *Int. J. Hum. Comput. Stud.*, vol. 49, no. 2, pp. 181–200, Aug. 1998.
- [45] M. Doherty and R. Anderson, "Variation in scatterplot displays," *Behav. Res. Methods*, vol. 41, no. 1, pp. 55–60, 2009.
- [46] "Scatterplot," in *The Concise Encyclopedia of Statistics SE - 363*, Springer New York, 2008, p. 475.
- [47] Y. Guo and G. Wang, "Assessing CBOC Signal Quality Using Scatterplot," in *China Satellite Navigation Conference (CSNC) 2012 Proceedings SE - 15*, vol. 161, J. Sun, J. Liu, Y. Yang, and S. Fan, Eds. Springer Berlin Heidelberg, 2012, pp. 169–178.

Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

- [48] R. Muenchen, "Traditional Graphics graphics graphics traditional," in *R for SAS and SPSS Users SE - 21*, Springer New York, 2009, pp. 281–339.
- [49] A. Awasthi and S. S. Chauhan, "A hybrid approach integrating Affinity Diagram, AHP and fuzzy TOPSIS for sustainable city logistics planning," *Appl. Math. Model.*, vol. 36, no. 2, pp. 573–584, Feb. 2012.
- [50] F. Geyer, U. Pfeil, J. Budzinski, A. Höchtl, and H. Reiterer, "AffinityTable - A Hybrid Surface for Supporting Affinity Diagramming," in *Human-Computer Interaction – INTERACT 2011 SE - 33*, vol. 6948, P. Campos, N. Graham, J. Jorge, N. Nunes, P. Palanque, and M. Winckler, Eds. Springer Berlin Heidelberg, 2011, pp. 477–484.
- [51] W. Widjaja, K. Yoshii, K. Haga, and M. Takahashi, "Discusys: Multiple User Real-time Digital Sticky-note Affinity-diagram Brainstorming System," *Procedia Comput. Sci.*, vol. 22, no. 0, pp. 113–122, 2013.
- [52] K. Holtzblatt, J. B. Wendell, and S. Wood, "Chapter 8 - Building an Affinity Diagram," in *Interactive Technologies*, K. Holtzblatt, J. B. Wendell, and S. B. T.-R. C. D. Wood, Eds. San Francisco: Morgan Kaufmann, 2005, pp. 159–179.
- [53] D. Mohamedally and P. Zaphiris, "Categorization Constructionist Assessment with Software-Based Affinity Diagramming.," *Int. J. Hum. Comput. Interact.*, vol. 25, no. 1, pp. 22–48, Jan. 2009.
- [54] H.-C. Liu, J.-X. You, X.-J. Fan, and Q.-L. Lin, "Failure mode and effects analysis using D numbers and grey relational projection method," *Expert Syst. Appl.*, vol. 41, no. 10, pp. 4670–4679, Aug. 2014.
- [55] A. Pillay and J. Wang, "Modified failure mode and effects analysis using approximate reasoning," *Reliab. Eng. Syst. Saf.*, vol. 79, no. 1, pp. 69–85, Jan. 2003.
- [56] F. J. Pino, O. Pedreira, F. García, M. R. Luaces, and M. Piattini, "Using Scrum to guide the execution of software process improvement in small organizations," *J. Syst. Softw.*, vol. 83, no. 10, pp. 1662–1677, Oct. 2010.
- [57] Q. Li, "A novel Likert scale based on fuzzy sets theory," *Expert Syst. Appl.*, vol. 40, no. 5, pp. 1609–1618, Apr. 2013.