



Centro de Investigación en Matemáticas, A.C.

CIMAT

Método para aligerar procesos de software mediante la optimización en la selección de prácticas de Ingeniería de Software

T E S I S

Que para obtener el grado de
Maestro en Ingeniería de Software

P r e s e n t a

Juan José Miramontes Sandoval

Director de Tesis:

Dra. Mirna Ariadna Muñoz Mata

Codirector:

Dr. Jezreel Mejía Miranda

Zacatecas, Zac. Agosto de 2016

Agradecimientos

A mi familia por su apoyo incondicional en mi formación personal y académica.

A mis amigos y compañeros de la octava generación de la Maestría en Ingeniería de Software, por su gran compañerismo y apoyo.

A los docentes del Centro de Investigación en Matemáticas, Unidad Zacatecas por los conocimientos, consejos y ejemplos recibidos a lo largo de mi formación académica.

A las empresas participantes en los casos de estudio, por su valioso tiempo y disposición para utilizar y evaluar el método propuesto en el presente trabajo de tesis.

Agradecimientos especiales a mi tutora y directora de tesis la Dra. Mirna Muñoz por todo el apoyo recibido a lo largo de la maestría, y por ser una excelente guía para el desarrollo de la tesis.

Finalmente, agradezco al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico otorgado para la realización de los estudios de maestría y el desarrollo del presente trabajo de tesis. De igual manera, agradezco al Centro de Investigación en Matemáticas por los apoyos otorgados para estancias de investigación y participación en congresos.

Abstract

One of the most important concerns in software industry is the development of software products with the optimal use of resources, time and costs, in other words, an organization of software development needs to be efficient and to use software engineering practices that allow it to be efficient according to its particular needs.

A feasible way to achieve this is through implementing software process improvements allowing the organization to select the best practices that help it to become more efficient. In this research is proposed a method to lighten software processes that enable organizations, especially SMEs, to assess the activities of their software development processes, so that, the processes are optimized based on practices that add more value to their processes, and therefore lead them to be more efficient.

Keywords: Software process lightening, Software process improvement, Added value analysis, agile development practices.

Resumen

Una de las preocupaciones más importantes en la industria de software es el desarrollo de productos de software con el uso óptimo de recursos, tiempo y costes, es decir, una organización de software necesita ser eficiente y contar con prácticas de Ingeniería de Software que le permitan ser eficiente de acuerdo a sus características particulares.

Un camino viable para lograrlo, es implementar una mejora de procesos de software que le permita a la organización seleccionar las mejores prácticas que la apoyen a ser más eficiente. En este trabajo de investigación se propone un método para aligerar procesos de software que permite a las organizaciones, especialmente a las PyMEs, evaluar las actividades que realizan en sus procesos de desarrollo de software, de manera que éstas se vayan optimizando con base en las prácticas que aporten más valor a sus procesos, y por lo tanto, las conduzca a ser más eficientes.

Palabras clave: Aligeramiento de procesos de software, Mejora de proceso de software, Análisis del valor añadido, prácticas de desarrollo ágil.

Índice

Abstract.....	iii
Resumen.....	iv
Introducción	1
Capítulo 1. Antecedentes.....	3
1.1. Marco teórico.....	3
1.1.1. Procesos de Software	3
1.1.1.1. Definición de procesos de software	4
1.1.2. Mejora de Procesos de Software.....	5
1.1.3. Modelos de Mejora de Procesos de Software	6
1.1.3.1. CMMI	6
1.1.3.2. ISO/IEC 15504 (SPICE)	9
1.1.3.3. MoProSoft.....	11
1.1.3.4. ISO/IEC 29110.....	13
1.2. Planteamiento del problema.....	15
1.3. Objetivos.....	16
1.3.1. Objetivo General.....	16
1.3.2. Objetivos específicos	16
1.4. Hipótesis	16
1.5. Justificación	17
Capítulo 2. Estado del arte.....	18
2.1. Revisión sistemática.....	18
2.1.1. Planificación de la revisión sistemática	18
2.1.1.1. Identificación de la necesidad de la revisión.....	19
2.1.1.2. Preguntas de investigación.....	19
2.1.1.3. Cadenas de búsqueda	19
2.1.1.4. Fuentes de datos	20
2.1.2. Ejecución de la revisión sistemática	20
2.1.2.1. Criterios de inclusión y exclusión.....	20
2.1.2.2. Selección de estudios primarios	20
2.1.2.3. Extracción de datos	21
2.1.3. Reporte de resultados.....	22

2.1.3.1.	Marcos de trabajo, métodos y metodologías para aligerar procesos.....	22
2.1.3.2.	Procesos enfocados	23
2.1.3.3.	Estrategias para aligerar procesos de software.....	25
2.2.	Trabajos relacionados	26
2.2.1.	Comparativa de propuestas	28
2.2.2.	Características cubiertas por los trabajos relacionados	29
2.3.	Selección de tecnología.....	30
2.3.1.	Frameworks de desarrollo Web	30
2.3.1.1.	Laravel	31
2.3.1.2.	Django.....	31
2.3.1.3.	Ruby on Rails.....	32
2.3.1.4.	Comparativa de Frameworks	32
2.3.2.	Gestores de Bases de datos	34
2.3.2.1.	MySQL	34
2.3.2.2.	MariaDB	35
2.3.2.3.	PostgreSQL.....	35
2.3.2.4.	Comparativa de gestores de bases de datos.....	36
2.4.	Metodología de desarrollo	36
2.4.1.	UWE	36
2.4.2.	WebML.....	37
2.4.3.	OOHDM	38
2.4.4.	Comparación de metodologías.....	38
Capítulo 3.	Metodología para el desarrollo de la tesis.....	40
3.1.	Ejecución de la metodología para el desarrollo de Tesis	41
Capítulo 4.	Método para aligerar procesos	43
4.1.	Desarrollo del método.....	43
4.2.	Prerrequisitos para el uso del método	44
4.3.	Descripción de los pasos del método	45
4.3.1.	Paso 1: Modelado del proceso de la empresa.....	45
4.3.1.1.	Propósito	46
4.3.1.2.	Información de entrada	46
4.3.1.3.	Fases/actividades.....	46

4.3.1.4.	Roles	47
4.3.1.5.	Productos de trabajo.....	47
4.3.2.	Paso 2: Análisis del valor añadido	47
4.3.2.1.	Propósito	48
4.3.2.2.	Información de entrada	48
4.3.2.3.	Fases/actividades.....	48
4.3.2.4.	Roles	50
4.3.2.5.	Productos de trabajo.....	50
4.3.3.	Paso 3: Mapeo de prácticas formales a prácticas ágiles.....	51
4.3.3.1.	Propósito	51
4.3.3.2.	Información de entrada	52
4.3.3.3.	Fases/actividades.....	52
4.3.3.4.	Roles	53
4.3.3.5.	Productos de trabajo.....	53
4.3.4.	Paso 4: Sugerencia de técnicas y herramientas	53
4.3.4.1.	Propósito	53
4.3.4.2.	Información de entrada	53
4.3.4.3.	Fases/actividades.....	53
4.3.4.4.	Roles	54
4.3.4.5.	Productos de trabajo.....	54
4.3.5.	Paso 5: Rediseño del proceso.....	54
4.3.5.1.	Propósito	54
4.3.5.2.	Información de entrada	55
4.3.5.3.	Fases/actividades.....	55
4.3.5.4.	Roles	56
4.3.5.5.	Productos de trabajo.....	56
4.4.	Desarrollo de la herramienta.....	56
4.4.1.	Modelo de Requerimientos	57
4.4.2.	Modelo de Contenido.....	58
4.4.3.	Modelo de Navegación	59
Capítulo 5.	Resultados.....	60
5.1.	Herramienta para implementación del método propuesto	60

5.2.	Casos de estudio.....	64
5.2.1.	Diseño y planificación de los casos de estudio	64
5.2.1.1.	Objetivo de los casos de estudio	65
5.2.1.2.	Objeto de estudio	65
5.2.1.3.	Marco de referencia	65
5.2.1.4.	Preguntas de investigación.....	66
5.2.1.5.	Métodos para la recogida de datos	66
5.2.2.	Preparación de la recogida de datos.....	66
5.2.3.	Recogida de datos	67
5.2.4.	Análisis de los datos recogidos	68
5.2.4.1.	Caso de estudio 1	68
5.2.4.2.	Caso de estudio 2	75
5.2.5.	Reporte de resultados	83
Capítulo 6.	Conclusiones	85
6.1.	Conclusiones	85
6.2.	Trabajo futuro	86
6.3.	Logros académicos.....	87
6.3.1.	Productos académicos	87
6.3.2.	Ponencias en congresos.....	87
6.3.3.	Estancias de Investigación	87
Referencias	89
Anexo A.	Estudios primarios de la revisión sistemática.....	92
Anexo B.	Tipo de propuestas y objetivos de los estudios primarios	93
Anexo C.	Cuestionario de prácticas específicas de Planificación de Proyectos	95
Anexo D.	Mapeo de prácticas del área de Planificación de Proyectos a prácticas de Scrum ...	98
Anexo E.	Autorización de publicación en formato electrónico de tesis	103

Índice de Figuras

Figura 1. Proceso como unión de Personas, procedimientos y tecnología (O'Regan, 2011).	4
Figura 2. Procesos en MoProSoft	13
Figura 3. Factores de éxito en proyectos de software	17
Figura 4. Proceso de selección de estudios primarios.....	21
Figura 5. Distribución de los estudios primarios por país.....	22
Figura 6. Número de estudios por tipo de propuesta	23
Figura 7. Número de estudios por área de procesos, organizados por categorías de CMMI.....	24
Figura 8. Metodología para el desarrollo de la tesis.	40
Figura 9. Propuesta del método para aligerar procesos	44
Figura 10. Acción a ejecutar de acuerdo al valor de la práctica.	50
Figura 11. Áreas de Proceso de la categoría de Gestión de Proyectos de CMMI cubiertas por Scrum (Marçal et al., 2007).....	51
Figura 12. Diagrama de casos de uso.....	57
Figura 13. Diagrama de clases.	58
Figura 14. Modelo de navegación.....	59
Figura 15. Interfaz de inicio (antes de registrarse o iniciar sesión).	60
Figura 16. Interfaz para el registro de cuentas de usuario.	61
Figura 17. Interfaz para agregar una nueva empresa..	61
Figura 18. Interfaz de inicio (después de iniciar sesión).....	62
Figura 19. Interfaz para la identificación de prácticas y captura del valor añadido.....	63
Figura 20. Herramienta para modelar el proceso de la empresa.	63
Figura 21. Interfaz con los resultados de la herramienta.	64
Figura 22. Pasos para implementar un caso de estudio.....	64
Figura 23. Análisis del valor añadido (Empresa A).....	69
Figura 24. Mapeo de prácticas formales a ágiles (Empresa A).....	69
Figura 25. Sugerencia de técnicas y herramientas (Empresa A).....	70
Figura 26. Proceso rediseñado (Empresa A).....	70
Figura 27. Respuestas de los empleados de la Empresa A para la pregunta A.....	71
Figura 28. Respuestas de los empleados de la Empresa A para la pregunta B.	71
Figura 29. Respuestas de los empleados de la Empresa A para la pregunta C.	72
Figura 30. Respuestas de los empleados de la Empresa A para la pregunta D.....	72
Figura 31. Respuestas de los empleados de la Empresa A para la pregunta E.	73
Figura 32. Respuestas de los empleados de la Empresa A para la pregunta F.	73
Figura 33. Respuestas de los empleados de la Empresa A para la pregunta G.....	74
Figura 34. Respuestas de los empleados de la Empresa A para la pregunta H.....	74
Figura 35. Respuestas de los empleados de la Empresa A para la pregunta I.	75
Figura 36. Análisis del valor añadido (Empresa B).	76
Figura 37. Mapeo de prácticas formales a ágiles (Empresa B).....	77
Figura 38. Mapeo de prácticas formales a ágiles (Empresa B).....	77

Figura 39. Proceso rediseñado (Empresa B).....	78
Figura 40. Respuestas de los empleados de la Empresa B para la pregunta A.	78
Figura 41. Respuestas de los empleados de la Empresa B para la pregunta B.	79
Figura 42. Respuestas de los empleados de la Empresa B para la pregunta C.	79
Figura 43. Respuestas de los empleados de la Empresa B para la pregunta D.	80
Figura 44. Respuestas de los empleados de la Empresa B para la pregunta E.	80
Figura 45. Respuestas de los empleados de la Empresa B para la pregunta F.....	81
Figura 46. Respuestas de los empleados de la Empresa B para la pregunta G.	81
Figura 47. Respuestas de los empleados de la Empresa B para la pregunta H.	82
Figura 48. Respuestas de los empleados de la Empresa B para la pregunta I.....	83

Índice de Tablas

Tabla 1. Comparación de los niveles de capacidad y madurez.....	7
Tabla 2. Áreas de proceso, categorías y niveles de madurez en CMMI-DEV.....	8
Tabla 3. Niveles y atributos en ISO/IEC 15504 SPICE.....	9
Tabla 4. Descripción de categorías del ISO/IEC 15504	10
Tabla 5. Procesos de ISO/IEC 15504 agrupados por categorías.....	10
Tabla 6. Estructura del estándar NMX-I-059-NYCE-2005	12
Tabla 7. Niveles de capacidad de MoProSoft y sus respectivos atributos de procesos	12
Tabla 8. Estructura del estándar ISO/IEC 29110.....	14
Tabla 9. Perfiles del grupo genérico en el ISO/IEC 29110.....	14
Tabla 10. Palabras clave, términos relacionados y cadena de búsqueda	19
Tabla 11. Número de estudios por área de procesos.....	24
Tabla 12. Estrategias para el aligeramiento de procesos	25
Tabla 13. Comparativa de propuestas	28
Tabla 14. Características cubiertas por los trabajos relacionados.....	30
Tabla 15. Características necesarias para el desarrollo de la herramienta	34
Tabla 16. Comparativa de gestores de bases de datos	36
Tabla 17. Modelos propuestos en las metodologías	38
Tabla 18. Roles para el paso 1: Modelado del proceso de la empresa.....	47
Tabla 19. Rangos para el análisis del valor añadido.....	49
Tabla 20. Roles para el paso 2: Análisis del valor añadido	50
Tabla 21. Mapeo de prácticas específicas a prácticas ágiles.....	52
Tabla 22. Sugerencia de técnicas y herramientas	54
Tabla 23. Roles para el paso 2: Análisis del valor añadido	56
Tabla 24. Descripción de casos de uso	57
Tabla 25. Empresas estudiadas	65
Tabla 26. Preguntas de la encuesta	67
Tabla 27. Datos sobre la Empresa A.....	68
Tabla 28. Datos sobre la Empresa B.....	75
Tabla 29. Criterios para definir el estado del mapeo de prácticas	98
Tabla 30. Mapeo de prácticas específicas a prácticas de Scrum (Objetivo específico 1)	98
Tabla 31. Mapeo de prácticas específicas a prácticas de Scrum (Objetivo específico 2)	99
Tabla 32. Mapeo de prácticas específicas a prácticas de Scrum (Objetivo específico 2)	101

Introducción

La mejora de procesos de software es un esfuerzo planificado, gestionado y controlado que tiene como objetivo mejorar la capacidad de los procesos de desarrollo de software de una organización (Pino, Pedreira, García, Luaces, & Piattini, 2010). Para ayudar a las organizaciones a lograr una mejora en sus procesos se han propuesto varios modelos y estándares, tales como: CMMI, ISO/IEC 15504, MoProSoft, entre otros, los cuales adoptan un enfoque basado en un conjunto de mejores prácticas que han tenido éxito en organizaciones de software (Pettersson, Ivarsson, Gorschek, & Öhman, 2008).

Sin embargo, los modelos y estándares de mejoras de procesos, en muchas ocasiones no son comprendidos de manera adecuada por las organizaciones, percibiéndose como “heavyweight”: demasiado grandes para comprender y poner en práctica (Kuilboer & Ashrafi, 2000; Reifer, 2000) debido a que demandan una gran cantidad de recursos y compromiso a corto y largo plazo. Esto hace que sea difícil para las organizaciones, en particular para las pequeñas y medianas empresas (PyMEs), iniciar y llevar a cabo actividades de mejora.

Por esta razón, las PyMEs implementan sólo un subconjunto de prácticas que les aporten valor a la organización y les ayuden a ser más eficientes, en lugar de implementar modelos o estándares completos.

Así mismo, las organizaciones que cuentan con procesos definidos basados en la implementación de modelos como CMMI y han adoptado una cultura de mejora continua, un camino factible para lograr esta mejora es la optimización de sus procesos a través de su aligeramiento.

Este trabajo presenta el desarrollo de un método que permite el aligeramiento de procesos mediante la optimización en la selección de prácticas en Ingeniería de Software. El método propuesto en este trabajo de investigación tiene su base en tres estrategias: (1) identificación de las mejores prácticas de la organización, (2) uso de principios Lean y (3) una combinación de prácticas formales y prácticas ágiles. Además se ha desarrollado una herramienta para facilitar el uso del método.

En este trabajo se presenta además la validación del método y la herramienta mediante la ejecución de dos casos de estudio en empresas con entornos completamente diferentes en cuanto experiencia en el uso de procesos, obteniendo en ambos casos resultados favorables.

La estructura de la tesis está compuesta por seis capítulos, los cuales se detallan a continuación:

Capítulo 1. Antecedentes. En este capítulo se describen conceptos fundamentales para la investigación dentro del marco teórico, se muestran también la problemática, los objetivos y la justificación.

Capítulo 2. Estado del arte. Muestra el estado actual con respecto al aligeramiento de procesos de software mediante una revisión sistemática de la literatura y un análisis de trabajos relacionados.

Capítulo 3. Metodología para el desarrollo de la tesis. Describe las tareas realizadas para el desarrollo de la tesis y el orden en el que se implementaron.

Capítulo 4. Método para aligerar procesos. En este capítulo se describe la propuesta del método para aligerar procesos, incluyendo una descripción detallada de cada uno de sus pasos, también se muestra en este capítulo el desarrollo de la herramienta para facilitar el uso el método propuesto.

Capítulo 5. Resultados. Describe el funcionamiento de la herramienta, posteriormente, detalla el diseño y análisis de los casos de estudio.

Capítulo 6. Conclusiones. Muestra algunas de las conclusiones generales del trabajo de tesis.

Capítulo 1. Antecedentes

En este capítulo se presentan los antecedentes, comenzando con el marco teórico que contiene la definición de los conceptos fundamentales a partir de los cuales se desarrolla la presente investigación, posteriormente se presenta: el planteamiento del problema, los objetivos generales y específicos y la justificación.

1.1. Marco teórico

El objetivo del marco teórico es brindar un marco de referencia conceptual necesario para comprender la investigación, el marco teórico se integra a partir de teorías, enfoques teóricos, estudios y antecedentes en general, que se consideren válidos para la adecuada fundamentación del estudio (Sampieri, Fernández, & Baptista, 2010). A continuación se presentan los conceptos fundamentales directamente involucrados con el desarrollo de la presente investigación.

1.1.1. Procesos de Software

El SWEBOK (Software Engineering Body of Knowledge) define a un proceso de software como un conjunto de actividades y tareas interrelacionadas que transforman los productos de trabajo de entrada en productos de trabajo de salida (IEEE Computer Society, 2014).

Para O'Regan (2011) un proceso de software es un conjunto de actividades, métodos, prácticas y transformaciones que las personas utilizan para desarrollar y mantener software y sus productos de trabajo asociados. En otras palabras, un proceso de software es el proceso utilizado por los ingenieros de software para diseñar y desarrollar programas informáticos. Puede ser un proceso no documentado e ideado para un proyecto en particular, o puede ser un proceso estandarizado y documentado utilizado por varios equipos en proyectos similares. Además de las definiciones anteriores, un proceso de software es visto como el adhesivo que une a las personas, procedimientos y herramientas o tecnología (Ver Figura 1).

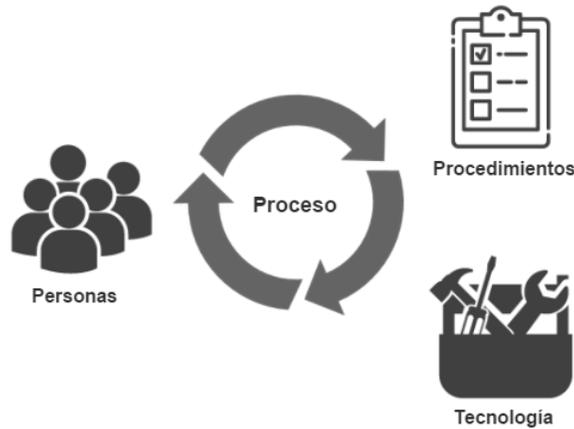


Figura 1. Proceso como unión de Personas, procedimientos y tecnología (O’Regan, 2011).

Los procesos de software son un factor crítico para la entrega de productos de software de calidad, ya que tienen como objetivo gestionar y transformar las necesidades del usuario en un producto de software que cumpla con sus requisitos (Barreto, Nunes, Rocha, & Murta, 2010).

La especificación de procesos de software es muy importante por las siguientes razones (IEEE Computer Society, 2014):

- Facilitan la comprensión humana, la comunicación y la coordinación
- Ayudan a la gestión de proyectos de software.
- Permiten medir y mejorar la calidad de los productos de software de manera eficiente.
- Apoyan la mejora de procesos.
- Proporcionan una base para el apoyo automático de la ejecución del proceso.

1.1.1.1. Definición de procesos de software

Se denomina definición del proceso de software a la descripción del proceso que se sigue para elaborar software. Cuando el proceso está diseñado y documentado de manera apropiada, la definición del proceso es la que guía a los ingenieros software sobre la manera de trabajar (Cuevas, 2002). Con una definición diseñada de forma adecuada se asegura que cada elemento de trabajo está asignado de modo apropiado y se conoce su estado en cada momento durante el ciclo de desarrollo de software.

El proceso seguido para elaborar software contiene una gran cantidad de actividades como por ejemplo: gestión de requisitos, planificación, gestión de configuración, gestión de calidad. Para cada una de estas actividades se puede definir un proceso de software.

En la definición de un proceso se deben identificar los siguientes componentes (Cuevas, 2002):

1. Roles o participantes en el proceso y sus responsabilidades.
2. Criterios de entrada, que son los elementos o condiciones necesarios para comenzar un proceso.
3. Tareas, pasos o actividades necesarias para transformar las entradas en salidas.
4. Salidas, datos o material producido o modificado por el proceso.
5. Criterios de salida, elementos o condiciones necesarias para establecer que se han obtenido las salidas.
6. Medidas de proceso, son las medidas que se recogerán mientras se realicen las actividades.

1.1.2. **Mejora de Procesos de Software**

En la actualidad muchas organizaciones de software tienen problemas al entregar software de alta calidad a sus clientes en el tiempo acordado. Hay problemas con sobrecostos en el presupuesto, entrega tardía del software, software entregado con poca calidad, quejas de los clientes con el funcionamiento del software, y baja moral en el personal de la organización (O'Regan, 2011). La mejora de procesos de software puede ayudar a reducir o eliminar estos problemas.

La mejora de procesos de software (Software Process Improvement, por sus siglas en inglés: SPI) tiene como objetivo hacer más eficientes a los procesos de software y aumentar la calidad de los productos mediante la evaluación continua y ajuste de los procesos (Petersen & Wohlin, 2010). Para Krasner (citado en: Pino, Pedreira, García, Luaces, & Piattini, 2010), la SPI es un esfuerzo planificado, gestionado y controlado que tiene como objetivo mejorar la capacidad de los procesos de desarrollo de software de una organización.

Por otro lado, O'Regan define a la SPI como un programa de actividades diseñadas para mejorar el rendimiento y la madurez de los procesos de software de la organización y los resultados de dicho programa.

Una organización que implementa la SPI obtiene los siguientes beneficios (O'Regan, 2011):

- Mejoras en la calidad
- Reducciones en el costo de mala calidad
- Mejoras en la productividad
- Reducciones en el costo de desarrollo de software
- Mejoras en la entrega a tiempo
- Mejoras en la coherencia en el presupuesto y calendario de entrega

- Mejoras en la satisfacción del cliente
- Mejoras en la moral de los empleados

1.1.3. Modelos de Mejora de Procesos de Software

Con el objetivo de lograr la mejora de los procesos de software, varios modelos y estándares han sido propuestos, tales como: el Capability Maturity Model Integration (CMMI) y el ISO 15504, MoProSoft y el ISO/IEC 29110. A continuación se da una breve descripción, estructura y los procesos que componen a cada uno de dichos modelos.

1.1.3.1. CMMI

A. Descripción

CMMI es un modelo de madurez y capacidad integrado, el cual contiene una colección de buenas prácticas que ayudan a las organizaciones a mejorar sus procesos. Aborda las prácticas que cubren el ciclo de vida del producto desde la concepción hasta la entrega y el mantenimiento (CMMI Product Team, 2010).

El principal objetivo de CMMI es reducir el costo en la implementación de mejoras en los procesos, eliminando inconsistencias y establecido guías para ayudar a las organizaciones en las distintas etapas de un proyecto de software como: planificación, gestión, entre otras (Selleri Silva et al., 2015).

CMMI fue desarrollado por el Software Engineerig Institute (SEI) de la Carnegie Mellon University (CMU), basado en algunos modelos de madurez y capacidad emergentes tales como: SW-CMM (Capability Maturity Model for Software), SECM (Systems Engineering Capability Model) e IPD-CMM (Integrated Product Development Capability Maturity Model), la primera versión de CMMI fue lanzada en el 2000 (Al-Tarawneh, Abdullah, & Ali, 2011). Actualmente CMMI se encuentra en su versión 1.3, publicada en noviembre de 2010 y está conformado por tres constelaciones: CMMI para Adquisición, CMMI para Servicios y CMMI para Desarrollo.

B. Estructura

La mejora de procesos en CMMI se puede lograr mediante dos caminos: el primero consiste en mejorar de forma incremental los procesos que corresponden a un área de proceso individual seleccionada por la organización, mientras que en el segundo las organizaciones mejoran un conjunto de procesos relacionados. Estos dos caminos se asocian con dos tipos de niveles: niveles de capacidad y niveles de madurez, respectivamente. Para alcanzar un nivel

particular, una organización debe satisfacer todas las metas del área de proceso que son objeto de la mejora, independientemente si es un nivel de capacidad o madurez (CMMI Product Team, 2010).

Los niveles de capacidad son llamados representación continua, debido a que se refieren a la consecución de la mejora de procesos en áreas de proceso individuales, hay cuatro niveles de representación continua, numerados del 0 al 3. Por otro lado, los niveles de madurez son llamados representación por etapas, ya que se refieren a la consecución de la mejora de proceso en múltiples áreas de proceso, un conjunto de áreas de proceso corresponde a un nivel de madurez, los cinco niveles de madurez se numeran del 1 al 5. La Tabla 1 muestra una comparación entre los niveles de capacidad y madurez (CMMI Product Team, 2010).

Tabla 1. Comparación de los niveles de capacidad y madurez

Nivel	Niveles de capacidad Representación continua	Niveles de madurez Representación por etapas
Nivel 0	Incompleto	
Nivel 1	Realizado	Inicial
Nivel 2	Gestionado	Gestionado
Nivel 3	Definido	Definido
Nivel 4		Gestionado cuantitativamente
Nivel 5		En optimización

C. Procesos

CMMI para Desarrollo (CMMI-DEV) contiene prácticas que cubren la gestión de proyectos, gestión de procesos, la ingeniería de sistemas, la ingeniería de hardware, la ingeniería de software y otros procesos de soporte utilizados en el desarrollo y mantenimiento. Se compone de 22 áreas de proceso, de las cuales, 16 son áreas de proceso base, 1 es un área de proceso compartida y 5 son áreas de proceso específicas de desarrollo (CMMI Product Team, 2010).

Cada área de proceso es definida en términos de los elementos del proceso: propósito, metas específicas (componente requerido) y prácticas específicas (componente esperado). Un componente requerido describe lo que una organización debe alcanzar para satisfacer un área de proceso; un componente esperado describe lo que una organización puede implementar para lograr un componente requerido (Pino, Baldassare, Piattini, & Visaggio, 2009).

Las áreas de proceso se organizan en cuatro categorías: Gestión de Procesos, Gestión de Proyectos, Ingeniería y Soporte. Estas categorías hacen hincapié en algunas de las relaciones clave que existen entre las áreas de proceso.

La Tabla 2 lista las 22 áreas de proceso ordenadas alfabéticamente de acuerdo a su acrónimo en inglés, en ella se muestran las categorías en la que se clasifican y el nivel de madurez al que pertenecen (CMMI Product Team, 2010).

Tabla 2. Áreas de proceso, categorías y niveles de madurez en CMMI-DEV

Área de proceso	Acrónimo	Categoría	Nivel de Madurez
Análisis Causal y Resolución	CAR	Soporte	5
Gestión de Configuración	CM	Soporte	2
Análisis de Decisiones y Resolución	DAR	Soporte	3
Gestión Integrada del Proyecto	IPM	Gestión de Proyectos	3
Medición y Análisis	MA	Soporte	2
Definición de Procesos de la Organización	OPD	Gestión de Procesos	3
Enfoque en Procesos de la Organización	OPF	Gestión de Procesos	3
Gestión del Rendimiento de la Organización	OPM	Gestión de Procesos	5
Rendimiento de Procesos de la Organización	OPP	Gestión de Procesos	4
Formación en la Organización	OT	Gestión de Procesos	3
Integración del Producto	PI	Ingeniería	3
Monitorización y Control Del Proyecto	PMC	Gestión de Proyectos	2
Planificación del Proyecto	PP	Gestión de proyectos	2
Aseguramiento de la Calidad del Proceso y del Producto	PPQA	Soporte	2
Gestión Cuantitativa del Proyecto	QPM	Gestión de Proyectos	4
desarrollo de requisitos	RD	Ingeniería	3
Gestión de requisitos	REQM	Gestión de proyectos	2
Gestión de riesgos	RSKM	Gestión de proyectos	3
Gestión de acuerdos con Proveedores	SAM	Gestión de proyectos	2
Solución Técnica	TS	Ingeniería	3
Validación	VAL	Ingeniería	3
Verificación	VER	Ingeniería	3

1.1.3.2. ISO/IEC 15504 (SPICE)

A. Descripción

El ISO/IEC 15504 también conocido como SPICE (Software Process Improvement and Capability Determination) es un estándar internacional para la evaluación y mejora de procesos de desarrollo de software (Ehsan, Perwaiz, Arif, Mirza, & Ishaque, 2010).

El modelo contenido en el estándar ISO 15504 describe los procesos que una organización puede realizar para adquirir, suministrar, desarrollar, explotar, evolucionar y soportar software y las prácticas genéricas que caracterizan la capacidad de esos procesos. El estándar ISO 15504 se basa en la descripción del estándar ISO 12207 de ciclo de vida para definir los procesos de una organización (Cuevas, 2002).

ISO/IEC 15504 proporciona una guía para la realización de evaluaciones, incluyendo el proceso, el modelo y las herramientas de evaluación. Inicialmente se desarrolló a partir de la fusión de CMMI, TRillium y Bootstrap. SPICE complementa enfoques de evaluación existentes como CMMI e ISO 9000. Puede ser utilizado para la evaluación del proceso como la determinación de la capacidad. Abarca todos los elementos importantes en el área de TI (Tecnologías de la Información) y desarrollo de software (Ehsan et al., 2010).

El estándar ISO/IEC 15504 está dividido en 5 partes, bajo el título general: Information technology — Process assessment:

- Part 1: Concepts and vocabulary
- Part 2: Performing an assessment
- Part 3: Guidance on performing an assessment
- Part 4: Guidance on use for process improvement and process capability determination
- Part 5: An exemplar Process Assessment Model

B. Estructura

El estándar ISO / IEC 15504 es similar a CMMI, se compone de niveles de madurez y cada nivel de madurez tiene atributos asignados a cada nivel (como se muestra en la Tabla 3).

Tabla 3. Niveles y atributos en ISO/IEC 15504 SPICE

Núm.	Nivel	Atributos
1	Realizado	- Rendimiento del proceso
2	Gestionado	- Gestión del desempeño - Gestión de productos de trabajo

3	Establecido	- Definición de procesos - Despliegue de procesos
4	Predecible	- Medición de procesos - Control de procesos
5	En optimización	- Innovación de procesos - Optimización continua

En el estándar ISO/IEC 15504 cada nivel de madurez se compone de un conjunto de procesos que contienen prácticas específicas (o base) y prácticas genéricas. Sin embargo, a diferencia de CMMI donde se describen las prácticas específicas y genéricas dentro de las áreas de proceso que se presentan en el mismo modelo, en ISO/IEC 15504 las prácticas genéricas se describen en los atributos del proceso presentados en la Parte 2 (ISO, 2004) [y se extienden en la Parte 5 (ISO, 2012)] de este estándar, y las prácticas base son descritas en los procesos presentados en el ISO/IEC 12207 (ISO, 2013) [o en ISO/IEC 15504-5 (ISO, 2012)].

Los procesos en ISO/IEC 15504 se agrupan en las categorías: cliente/proveedor, ingeniería, soporte, gestión, y organización (Tabla 4).

Tabla 4. Descripción de categorías del ISO/IEC 15504

Categoría de proceso	Descripción
Cliente-Suministrador	Procesos que afectan directamente al cliente.
Ingeniería	Procesos que especifican, implementan o mantienen un sistema y el producto de software.
Proyecto	Procesos que establecen el proyecto, coordinan y gestionan sus recursos.
Soporte	Procesos que permiten y soportan la ejecución de otros procesos en un proyecto.
Organización	Procesos que establecen los objetivos de negocio de la organización y el proceso de desarrollo, el producto y el valor de los recursos.

C. Procesos

El estándar ISO/IEC 15504 consta de 25 procesos organizados en 5 categorías: Procesos de Acuerdos (AGR), Procesos Empresariales (ENT), Procesos para Proyectos (PRJ), Procesos Técnicos (TEC) y Procesos para Adaptación (TRL). A continuación en la Tabla 5 se muestra la lista de procesos en ISO/IEC 15504 organizados por categoría.

Tabla 5. Procesos de ISO/IEC 15504 agrupados por categorías

Clave	Procesos	Categoría
-------	----------	-----------

AGR.1	Proceso de Adquisición	Procesos de acuerdos (AGR)
AGR.2	Proceso de Suministro	Procesos de acuerdos (AGR)
ENT.1	Proceso de Gestión del Entorno Empresarial	Proceso empresariales (ENT)
ENT.2	Proceso de Gestión de Inversiones	Proceso empresariales (ENT)
ENT.3	Proceso de Gestión del Modelo Ciclo de Vida	Proceso empresariales (ENT)
ENT.4	Proceso de Gestión de Recursos Humanos	Proceso empresariales (ENT)
ENT.5	Proceso de Aseguramiento de la Calidad del Software	Proceso empresariales (ENT)
PRJ.1	Proceso de Planificación del Proyecto	Procesos del Proyecto (PRJ)
PRJ.2	Proceso de Evaluación y Control del Proyecto	Procesos del Proyecto (PRJ)
PRJ.3	Proceso de Control del Proyecto	Procesos del Proyecto (PRJ)
PRJ.4	Proceso de Gestión de la Decisión	Procesos del Proyecto (PRJ)
PRJ.5	Proceso de Gestión de Riesgos	Procesos del Proyecto (PRJ)
PRJ.6	Proceso de Gestión de la Configuración	Procesos del Proyecto (PRJ)
PRJ.7	Proceso de Gestión de la Información	Procesos del Proyecto (PRJ)
TEC.1	Proceso de Definición de Requisitos de los Stakeholders	Proceso Técnicos (TEC)
TEC.2	Proceso de Análisis de los Requisitos del Sistema	Proceso Técnicos (TEC)
TEC.3	Proceso de Diseño de la Arquitectura del Sistema	Proceso Técnicos (TEC)
TEC.4	Proceso de Implementación del Software	Proceso Técnicos (TEC)
TEC.5	Proceso de Integración del Sistema	Proceso Técnicos (TEC)
TEC.6	Proceso de Verificación del Software	Proceso Técnicos (TEC)
TEC.7	Proceso de Transición	Proceso Técnicos (TEC)
TEC.8	Proceso de Validación del Software	Proceso Técnicos (TEC)
TEC.9	Proceso de Operación del Software	Proceso Técnicos (TEC)
TEC.10	Proceso de Mantenimiento	Proceso Técnicos (TEC)
TRL.1	Proceso de Adaptación	Procesos de Adaptación (TLR)

1.1.3.3. MoProSoft

A. Descripción

MoProSoft (Modelo de Procesos para la Industria del Software) es un modelo de procesos diseñado en México como parte del programa PROSOFT de la Secretaría de Economía, dirigido entre otros objetivos a elevar el nivel de madurez de capacidades de las pequeñas y medianas empresas (PyMEs) dedicadas al desarrollo y mantenimiento de software (NYCE, 2011b). MoProSoft está basado en CMMI y los estándares ISO/IEC 9001:2000 e ISO/IEC 12207 (Rios, Vargas, Espinoza, & Peralta, 2008).

MoProSoft fomenta la estandarización de su operación, a través de la incorporación de las mejores prácticas en gestión en ingeniería de software. Mediante la adopción del modelo, una

organización de software puede elevar su capacidad para ofrecer servicios con calidad y alcanzar niveles internacionales de calidad (Silva, 2004).

El nombre oficial del estándar en conjunto con su modelo de evaluación de procesos (EvalProSoft) es NMX-I-059-NYCE-2005: Tecnología de la información - Software - Modelos de procesos y evaluación para desarrollo y mantenimiento de software.

El estándar NMX-I-059-NYCE-2005 se compone 4 partes que se muestran a continuación:

Tabla 6. Estructura del estándar NMX-I-059-NYCE-2005

Parte	Descripción
Parte 01: NMX-I-059/01-NYCE-2005	Definición de conceptos y productos
Parte 02: NMX-I-059/02-NYCE-2005	Requisitos y procesos (MoProSoft)
Parte 03: NMX-I-059/03-NYCE-2005	Guías para la implementación de procesos
Parte 04: NMX-I-059/04-NYCE-2005	Guías para la evaluación de procesos (EvalProSoft)

B. Estructura

Similar a CMMI e ISO/IEC 15504 MoProsoft cuenta con 6 niveles de capacidad numerados del 0 al 5 (Tabla 7), donde el nivel 0 está asociado con el menor nivel de capacidad indicando que el proceso no es alcanzado, mientras que el nivel 5 significa el logro de los objetivos de negocio planeados mediante la optimización y mejora continua de procesos.

Tabla 7. Niveles de capacidad de MoProSoft y sus respectivos atributos de procesos

Núm.	Nivel	Atributos
0	Incompleto	
1	Realizado	1.1 Realización del proceso
2	Gestionado	2.1 Gestión de la ejecución 2.2 Gestión de los productos
3	Establecido	3.1 Definición del proceso 3.2 Recursos del proceso
4	Predecible	4.1 Medida del proceso 4.2 Control del proceso
5	Optimizado	5.1 Cambio de proceso 5.2 Mejora continua

C. Procesos

MoProSoft se ha enfocado a procesos que se encuentran divididos de a tres categorías que representan los niveles de la organización: Alta dirección, Gerencia y Operación. La Figura 2 muestra los procesos de cada categoría. La categoría Alta dirección sólo contiene el proceso Gestión de negocio, mientras que la categoría Gerencia se divide en tres procesos: Gestión de procesos, Gestión de proyectos, y Gestión de recursos, ésta última a su vez se divide en tres subprocesos: Recursos humanos y ambiente de trabajo, Bienes servicios e infraestructura y Conocimiento de la Organización, la categorías de Operación se integra de dos procesos: Administración de proyectos específicos y Desarrollo y mantenimiento del software.

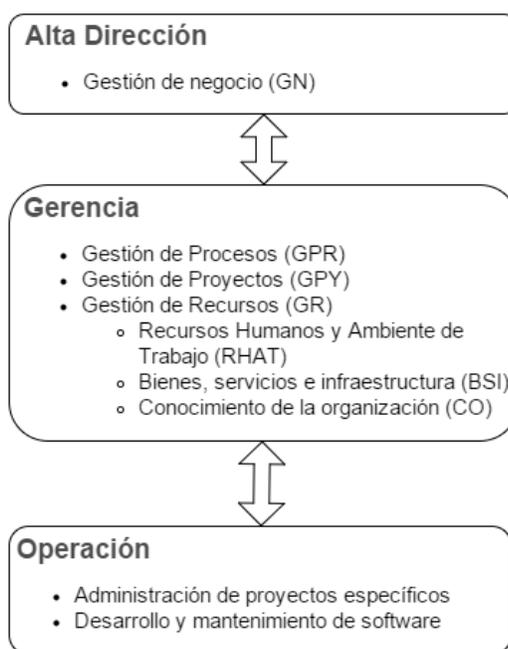


Figura 2. Procesos en MoProSoft

1.1.3.4. ISO/IEC 29110

A. Descripción

El estándar ISO/IEC 29110 está diseñado para empresas muy pequeñas (VSE, Very Small Entities) o microempresas, mismas que tienen una capacidad no mayor a 25 personas. Comprende un conjunto de normas y guías que se han desarrollado de acuerdo a las características y necesidades de las microempresas, basándose en subconjuntos de elementos de estándares apropiados de acuerdo al perfil de la microempresa (ISO, 2011).

La ISO/IEC 29110 se desarrolla con el objetivo de mejorar la calidad del producto y/o servicio de software, y para mejorar el desempeño de la organización, sin pretender excluir el

uso de diferentes metodologías de Ciclo de Vida tales como: Cascada, Iterativo, Incremental, Evolutivo o Ágil (NYCE, 2011a).

La ISO/IEC 29110 se divide en 5 partes de acuerdo al tipo de audiencia a la que está dirigida, las cuales se muestran a continuación:

Tabla 8. Estructura del estándar ISO/IEC 29110

ISO/IEC 29110	Título	Audiencia
Parte 1	Visión General	Empresas, evaluadores, desarrolladores, consultores, etc.
Parte 2	Marco de Referencia y Taxonomía	Normalizadores, desarrolladores, consultores.
Parte 3	Guía de Evaluación	Evaluadores y empresas.
Parte 4	Especificaciones de los Perfiles	Normalizadores, desarrolladores, consultores.
Parte 5	Guía de Gestión e Ingeniería	Empresas.

B. Estructura

Con el fin de adaptarse a las necesidades de la microempresas, el estándar ISO/IEC 29110 define perfiles, los cuales son un conjunto de uno o más estándares base y/o perfiles estandarizados para lograr una función en particular (Ramos & Mendoza, 2014). Hasta la fecha se han definido los perfiles que conforman el llamado Grupo Genérico. Este grupo es aplicable a microempresas que no desarrollan sistemas críticos. El grupo genérico se basa principalmente en la norma ISO / IEC 12207 (Procesos de ciclo de vida del software). Además, el grupo de trabajo ISO se apoya en el modelo mexicano MoProSoft para la elaboración del estándar ISO / IEC 29110 (NYCE, 2011a). Dicho grupo se compone de cuatro perfiles: de entrada, básico, intermedio y avanzado, los cuales se describen a continuación (Laporte, O'Connor, Hude, & Marvin, 2015):

Tabla 9. Perfiles del grupo genérico en el ISO/IEC 29110

Perfil	Descripción
Entrada	Dirigido a microempresas que trabajan en proyectos muy pequeños y para Startups. Ejemplo: máximo 6 meses de esfuerzo por persona.
Básico	Describe prácticas de desarrollo para una sola aplicación por un único equipo de proyecto.
Intermedio	Dirigido a microempresas que desarrollan varios proyectos con más de un equipo dentro de la organización.
Avanzado	Dirigido a microempresas que deseen crecer y mantenerse como como

Actualmente solo se encuentra publicado un conjunto de documentos para el perfil básico de ingeniería de software, los perfiles restantes se encuentran en preparación.

Perfil Básico

El perfil básico define dos procesos principales: Gestión del Proyecto (GP) e Implementación del software (IS). El propósito del proceso Gestión del Proyecto es establecer y llevar a cabo de manera sistemática las tareas del proyecto de implementación de software, las cuales permiten cumplir con los objetivos del proyecto en calidad, tiempo y costos esperados. Por su parte, el objetivo del proceso Implementación de software es la realización sistemática de las actividades de análisis, diseño, construcción, integración y pruebas para productos de software nuevos o modificados de acuerdo con los requisitos especificados (Ramos & Mendoza, 2014).

Ambos procesos están estrechamente relacionados y se ejecutan en paralelo. Se comienza con la planificación inicial de proyecto, misma con la que se realizará el monitoreo del proyecto. Posteriormente, durante la ejecución del proyecto se ejecutan las actividades de IS utilizando el plan del proyecto para evitar o reducir el impacto de las desviaciones (Ramos & Mendoza, 2014).

1.2. Planteamiento del problema

Una de las preocupaciones más importantes en la industria de software es el desarrollo de productos de software con el uso óptimo de recursos, tiempo y costes, es decir, una empresa de software necesita ser eficiente y contar con prácticas de ingeniería de software que le ayuden a tener un desarrollo de software óptimo (Garzás, Pino, Piattini, & Fernández, 2013).

Para que una organización pueda ser eficiente necesita adoptar una cultura de mejora continua, implementando prácticas de software que la apoyen para tal fin. Sin embargo, los modelos y estándares de mejora de procesos como CMMI e ISO/IEC 15504, no son comprendidos de manera adecuada, por lo que son vistos por muchos como “heavyweight”: demasiado grandes para comprender y poner en práctica (Kuilboer & Ashrafi, 2000; Reifer, 2000), debido a que demandan un gran cantidad de recursos y compromisos a corto y largo plazo.

Esto hace que sea difícil para las organizaciones, en particular para las pequeñas y medianas empresas (PyMEs), iniciar y llevar a cabo actividades de evaluación y mejora, considerando el tiempo y costo requerido (Calvo-Manzano Villalón et al., 2002; Kuilboer &

Ashrafi, 2000; Reifer, 2000). Por esta razón la mayoría de las PyMEs no implementan modelos completos, sino sólo las “prácticas” que consideran más aportan valor.

En organizaciones de software que cuentan con procesos definidos con base en la aplicación de modelos y estándares como CMMI e ISO/IEC 15504, que han adoptado una cultura de mejora continua, un camino factible para lograr dicha mejora es la optimización de sus procesos a través de aligeramiento. Citando la frase de Saint-Exupery (Kruchten, 2011): "La perfección se consigue, no cuando no haya más que añadir, sino cuando no hay nada más por quitar".

1.3. Objetivos

El objetivo general y los objetivos específicos para el desarrollo del presente trabajo de investigación se muestran a continuación:

1.3.1. Objetivo General

Desarrollar un método que permita aligerar procesos de software mediante la optimización en la selección de prácticas de Ingeniería de Software.

1.3.2. Objetivos específicos

- I. Obtener el estado del arte respecto al aligeramiento de procesos de software, centrado en tres aspectos clave: marcos de trabajo, métodos y metodologías utilizadas, procesos enfocados, y estrategias implementadas.
- II. Identificar cuáles son los modelos y estándares más utilizados en la industria de software.
- III. Implementar un catálogo de técnicas y herramientas.
- IV. Desarrollar un método para aligerar procesos de software.
- V. Diseñar y desarrollar una aplicación para facilitar el uso del método.
- VI. Validar el método y la aplicación mediante un caso de estudio.

1.4. Hipótesis

Si se dispone de un método y su herramienta de implementación dirigidos a la optimización en la selección de prácticas de ingeniería de software, entonces:

H1.1: Los empleados de una organización de software que utilicen el método perciben que los resultados obtenidos por el método pueden ayudar a incrementar la eficiencia del proceso de la organización.

1.5. Justificación

El aligeramiento de procesos de software es un camino que las empresas de desarrollo de software pueden seguir para hacer más eficientes sus procesos y lograr una cultura de mejora continua, además, tener procesos menos pesados les permite a las empresas satisfacer más rápido las necesidades de los clientes, con menores costos y conservando la calidad.

En el Manifiesto del Chaos 2013 (The Standish Group International, 2013) se mencionan los factores que le permiten a una organización desarrollar proyectos de software con éxito (en tiempo, presupuesto y con la calidad deseada). Entre estos factores, la optimización y los procesos ágiles son directamente abordados en la propuesta para aligerar procesos mostrada en el Capítulo 4 (ver Figura 3).

Factors of Success	Points
Executive management support	20
User involvement	15
Optimization	15
Skilled resources	13
Project management expertise	12
Agile process	10
Clear business objectives	6
Emotional maturity	5
Execution	3
Tools and infrastructure	1

Figura 3. Factores de éxito en proyectos de software

Durante los últimos 30 años se han llevado a cabo esfuerzos para implementar grandes modelos de procesos de software, sin embargo no se ha tenido un gran éxito en ello (Kruchten, 2011), por lo cual es necesario analizar porque no han funcionado y donde fallaron. En este contexto, se resalta la importancia de apoyar a las empresas en la implementación de prácticas que les apoyen a ser más eficientes.

Capítulo 2. Estado del arte

En este capítulo se presentan los resultados del método de la revisión sistemática cuyo objetivo es obtener el estado del arte respecto al aligeramiento de procesos, orientándose en tres aspectos principales: 1) marcos de trabajo, métodos y metodologías, 2) procesos enfocados y 3) estrategias para aligerar procesos. En la segunda parte del capítulo se analizan los trabajos con mayor similitud al desarrollado en la presente investigación. Este capítulo incluye también la selección de tecnología y la selección de la metodología para el desarrollo de la herramienta.

2.1. Revisión sistemática

La revisión sistemática de la literatura (Systematic Literature Review, por sus siglas en inglés SLR) es un método que permite identificar, evaluar e interpretar toda la evidencia disponible relevante a un tema o pregunta de investigación (Kitchenham & Charters, 2007). La SLR reduce la posibilidad de sesgo en las búsquedas de estudios, ya que es necesario definir un protocolo que especifica los métodos utilizados para guiar la revisión sistemática (Selleri Silva et al., 2015).

La revisión sistemática se constituye de tres fases principales: planificación de la revisión, ejecución de la revisión, y reporte de resultados. A continuación se detallan las actividades realizadas en cada fase.

2.1.1. Planificación de la revisión sistemática

La planificación es la primera fase de la revisión sistemática, comprende las siguientes actividades: confirmar la necesidad de la revisión, especificar las preguntas de investigación, enumerar las fuentes de datos para realizar las búsquedas y, formular las cadenas de búsqueda.

2.1.1.1. Identificación de la necesidad de la revisión

Una de las preocupaciones más importantes en la industria de software es el desarrollo de productos de software con el uso óptimo de recursos, tiempo y costos, esto es, una organización de software necesita ser eficiente y tener un óptimo desarrollo de software (Garzías et al., 2013). De acuerdo con Rizawan & Hussain (2008), para ser eficiente, una organización necesita aligerar sus procesos de software. La revisión sistemática se realiza por lo tanto, con el objetivo de conocer el estado actual del aligeramiento de procesos de software, enfocándose en tres elementos clave: marcos de trabajo, métodos y metodologías, procesos de software enfocados y estrategias utilizadas.

2.1.1.2. Preguntas de investigación

Las preguntas de investigación son las siguientes:

RQ1. ¿Qué **marcos de trabajo, métodos o metodologías** para **aligerar u optimizar procesos de software** existen?

RQ2. ¿Cuáles son los **procesos** que más se **aligeran**?

RQ3. ¿Qué estrategias son utilizadas para hacer más **aligerar** los **procesos de software**?

2.1.1.3. Cadenas de búsqueda

A partir de las preguntas de investigación se resaltan las palabras consideradas clave para realizar las búsquedas. En la Tabla 10 se describen sinónimos y términos relacionados (en inglés) con las palabras clave, y la cadena de búsqueda formulada uniendo las palabras clave mediante los conectores lógicos AND y OR.

Tabla 10. Palabras clave, términos relacionados y cadena de búsqueda

Palabras clave	Sinónimos / términos relacionados	Cadena de búsqueda
Marcos de trabajo, métodos y metodologías	Frameworks, methods, methodologies	
Lighten	Lightweight / light-weight Lean Light Lighthen	(framework OR method OR methodology) AND (lightweight OR lean OR lighten OR light OR optimize OR optimizing) AND software process
Optimizar	Optimize optimizing	
Procesos de software	Software process	

2.1.1.4. Fuentes de datos

Para la ejecución de la revisión sistemática se seleccionan las fuentes de datos consideradas relevantes en el área de Ingeniería de Software:

- IEEE Xplore
- Elsevier Science (Science Direct)
- SpringerLink
- ACM Digital Library

2.1.2. Ejecución de la revisión sistemática

La segunda fase de la SLR es la ejecución de la revisión, en la cual se definen los criterios de inclusión y exclusión para seleccionar los estudios primarios. Se muestra el proceso de selección de estudios primarios y el número de estudios encontrados en cada paso del proceso. Por último, se muestra la plantilla para extracción de datos de los estudios primarios.

2.1.2.1. Criterios de inclusión y exclusión

Criterios de inclusión:

1. Estudios en los idiomas inglés o español.
2. Estudios a partir del año 2008.
3. Estudios que contengan al menos dos palabras clave en el título y abstract.
4. Estudios que contengan análisis, evaluación o aplicaciones de modelos o estándares en procesos de software.
5. Estudios que contengan casos prácticos o validación de su propuesta.
6. Estudios que muestren resultados sobre la aligeración u optimización de procesos de software.

Criterios de exclusión:

1. Estudios que no contengan información sobre la aligeración de procesos de software o sobre procesos ligeros de software.
2. Estudios repetidos.
3. Estudios a los que no se tiene acceso.
4. Estudios que no cumplan con al menos 4 criterios de inclusión.

2.1.2.2. Selección de estudios primarios

La selección de estudios primario se lleva a cabo ejecutando un proceso de selección para la obtención de estudios primarios, el cual consiste en cinco pasos:

- Paso 1. Tomar la cadena y adaptarla a los motores de búsqueda de las fuentes de datos.
- Paso 2. Filtrar los estudios aplicando los primeros tres criterios de inclusión.
- Paso 3. Leer títulos y abstracts para identificar estudios potencialmente relevantes.
- Paso 4. Aplicar criterios de inclusión restantes, al leer introducción, métodos y conclusión, si es necesario, todo el estudio.
- Paso 5. Obtener los estudios primarios.

La Figura 4 muestra la aplicación de los pasos y el número de estudios obtenidos.

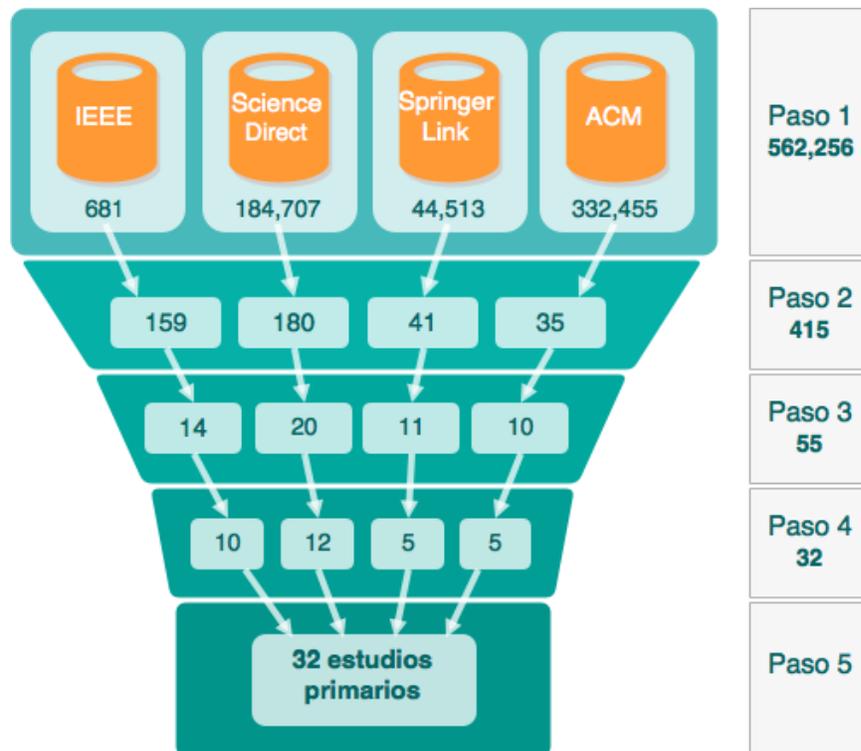


Figura 4. Proceso de selección de estudios primarios

Como se observa en la Figura 1, de los 562,256 estudios encontrados al ejecutar las cadenas, sólo 32 cumplieron con los criterios de inclusión y exclusión, los cuales fueron analizados y utilizados para la presente investigación.

2.1.2.3. Extracción de datos

Los datos extraídos de los estudios primarios se registraron en un editor de hojas de cálculo (Microsoft Excel™), en el cual se realiza una plantilla que contiene los siguientes datos: título, autores, año, palabras clave, fuente, objetivo general, problema, estrategia para aligerar procesos, forma en que se valida, conclusiones, fase o proceso que se aligera, tipo (marco de trabajo, método, metodología).

2.1.3. Reporte de resultados

En esta sección se hace un análisis de los resultados obtenidos de la extracción y análisis de datos de los estudios primarios con el objetivo de establecer el estado actual respecto a marcos de trabajo, métodos o metodologías utilizados en el aligeramiento de procesos de software, los procesos en los que se enfocan, y las estrategias que utilizan para lograrlo.

En la Figura 5 se muestra un gráfico geográfico de la distribución de estudios primarios por país. Como se observa en el gráfico, con 5 estudios Estados Unidos de América es el país en más investigación se han realizado sobre el tema, seguido por China con 4, en tercer lugar con 3 estudios se encuentran: Canadá, Finlandia y Suecia, seguidos por España y Malasia con sólo 2.

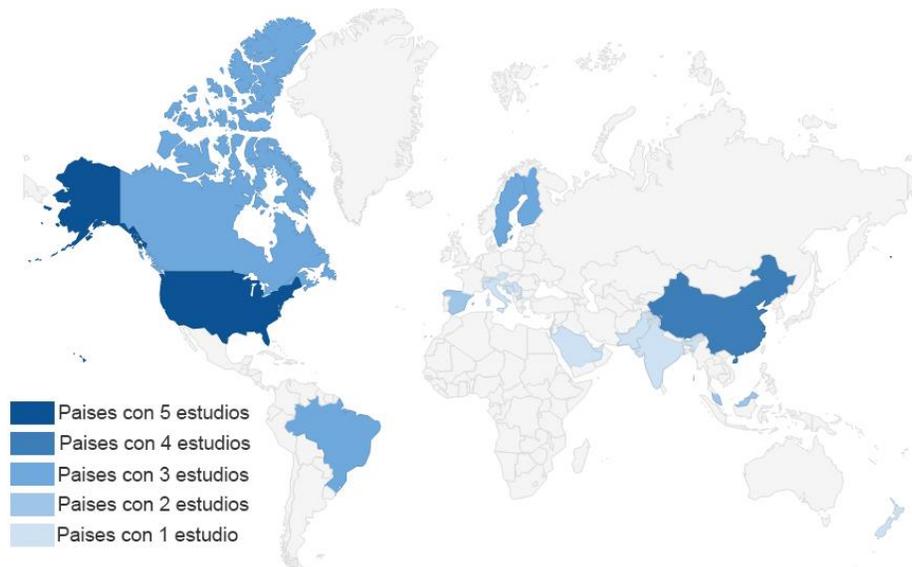


Figura 5. Distribución de los estudios primarios por país

2.1.3.1. Marcos de trabajo, métodos y metodologías para aligerar procesos

Con respecto a marcos de trabajo, métodos y metodologías para aligerar procesos de software, el Anexo B contiene un resumen de los objetivos de los estudios primarios, en donde se observa que tipo de propuesta presenta cada estudio (ya sea marco de trabajo, método o metodología). Además la Figura 6 muestra el número de estudios encontrados por tipo de propuesta, la suma de los estudios no corresponde a 32 (el número de estudios primarios) debido a que se encontraron estudios que abordan el tema del aligeramiento de procesos pero no presentan una propuesta.

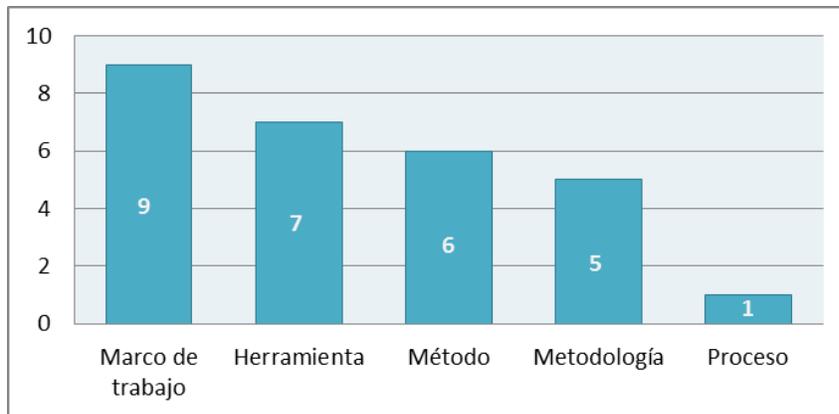


Figura 6. Número de estudios por tipo de propuesta

Aunque la cadena de búsqueda fue diseñada con el propósito de encontrar métodos, marcos de trabajo y metodologías, en los estudios también se encuentran otros tipos de propuesta: herramientas y procesos. Como se observa en la figura, la mayoría de los estudios están relacionados con marcos de trabajo y herramientas, seguidos por métodos y metodologías respectivamente.

2.1.3.2. **Procesos enfocados**

Durante la extracción de datos, se encuentran estudios enfocados tanto en áreas de proceso de CMMI como en fases de ciclo de vida de un proyecto, por lo cual se clasifican los procesos y fases de ciclo de vida en áreas de proceso de CMMI. En esta clasificación, los estudios que aligeran fases de diseño como la arquitectura de software son considerados como solución técnica, así como que los estudios que se enfocan en pruebas de software se clasificaron en validación o verificación según corresponda.

El número de estudios clasificados por áreas de procesos y niveles de madurez se muestra en la Tabla 11. Como se observa en la tabla, las áreas de proceso en las que más se enfocan los estudios son: solución técnica y validación, seguidas por el enfoque de procesos de la organización y el desarrollo de requisitos respectivamente. Además si se analiza por niveles de madurez, se observa también que nivel de madurez más cubierto es el 3, seguido por el nivel 2, mientras para los niveles 4 y 5 hay un escaso número de estudios.

Se realiza también una clasificación de los estudios en base a las categorías de áreas de proceso de CMMI (ver Figura 7), donde se observa que la categoría con más estudios es la de ingeniería seguida por la categoría de administración de proyectos.

Tabla 11. Número de estudios por área de procesos

Nivel de madurez 2	Abbr.	NE*	Nivel de Madurez 3	Abbr.	NE*
Gestión de Configuración	CM	2	Integración del Producto	PI	2
Medición y Análisis	MA	2	Gestión Integrada del Proyecto	IPM	1
Monitorización y Control del Proyecto	PMC	2	Definición de Procesos de la Organización	OPD	2
Aseguramiento de la Calidad del Proceso y del Producto	PPQA	2	Enfoque en Procesos de la Organización	OPF	4
Gestión de Requisitos	REQM	2	Formación en la Organización	OT	1
Gestión de Acuerdos con Proveedores	SAM	2	Análisis de Decisiones y Resolución	DAR	1
Planificación del Proyecto	PP	2	Desarrollo de Requisitos	RD	3
			Gestión de Riesgos	RSKM	1
			Solución Técnica	TS	5
			Validación	VAL	5
			Verificación	VER	2
	<i>Total</i>	<i>14</i>		<i>Total</i>	<i>27</i>
Nivel de madurez 4	Abbr.	NE*	Nivel de Madurez 5	Abbr.	NE*
Rendimiento de Procesos de la Organización	OPP	0	Gestión del Rendimiento de la Organización	OPM	0
Gestión Cuantitativa del Proyecto	QPM	0	Análisis Causal y Resolución	CAR	1
	<i>Total</i>	<i>0</i>		<i>Total</i>	<i>1</i>

*NE = Número de estudios

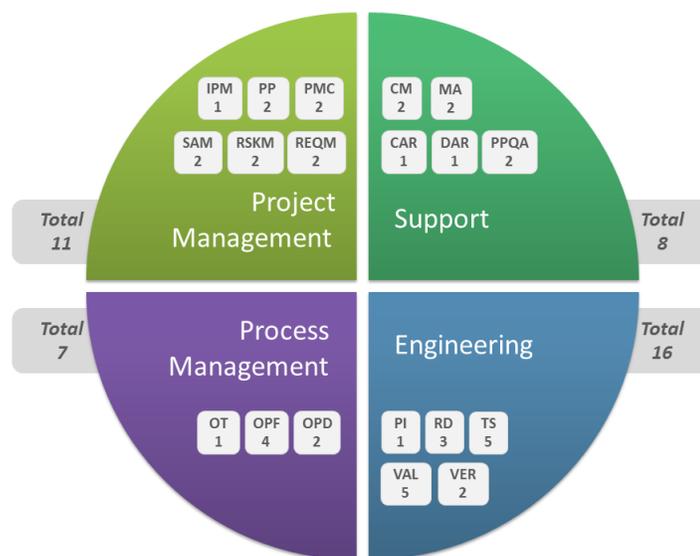


Figura 7. Número de estudios por área de procesos, organizados por categorías de CMMI

2.1.3.3. Estrategias para aligerar procesos de software

Para conocer las estrategias utilizadas en la aligeración de procesos, se realiza una categorización en base a la estrategia utilizada para aligerar procesos de software, encontrándose las siguientes estrategias: uso de herramientas, combinación de prácticas formales y ágiles, identificación de factores de éxito en el desarrollo ágil de software (ASD, Agile Software Development), identificación de las mejores prácticas de una organización, reuniones entre partes interesadas (stakeholders), diseño de metodologías y uso de procesos Lean. En la Tabla 12 se muestran las estrategias con una respectiva descripción y el número de estudios que las utilizan, el número es estudios no suma 32 (número de estudios primarios), porque algunos estudio no presentan estrategias.

Tabla 12. Estrategias para el aligeramiento de procesos

Estrategias	Descripción	NE*
Uso de herramientas	Utilizar herramientas para reducir el tiempo y esfuerzo utilizado en un proceso.	8
Combinación de prácticas formales y ágiles	Integrar prácticas de métodos formales con prácticas de desarrollo ágil.	6
Identificación de factores de éxito en ASD	Identificar elementos críticos de éxito en metodologías ágiles y modelando un proceso de software.	3
Identificación de las mejores prácticas de una organización	Identificar las prácticas clave que necesita una organización para cumplir sus objetivos de negocio, y basar su proceso en ellas.	3
Reuniones entre partes interesadas	Agilizar las salidas de un proceso mediante la realización de reuniones / workshops de partes interesadas.	3
Diseño de metodología	Diseñar una metodología ligera para procesos de software.	3
Uso de principios Lean.	Utilizar principios Lean: eliminar actividades que no son de valor añadido.	3

*NE = Número de estudios

La estrategia más utilizada para aligerar procesos de software es el uso de herramientas para automatizar proceso reduciendo el tiempo y esfuerzo que toma su implementación, seguida por la integración de prácticas de métodos formales con prácticas de desarrollo ágil de software. Así mismo se concluye que las estrategias para aligerar procesos de software identificadas no son excluyentes, es decir, pueden combinarse para establecer un enfoque de aligeración de procesos.

2.2. Trabajos relacionados

En esta sección se presentan los trabajos relacionados con la presente investigación, los cuales se toman como base para la realización de la propuesta del presente trabajo de investigación.

Los autores (Diaz, Garbajosa, & Calvo-Manzano, 2009) en su trabajo *Mapping CMMI Level 2 to Scrum Practices: An Experience Report* proponen introducir métodos ágiles como Scrum y dar cumplimiento a los procesos del modelo CMMI, esto debido a la problemática de que las estrategias de Mejora de Procesos de Software SPI basadas en CMMI-DEV requieren procesos de desarrollo de software pesados y grandes inversiones en términos de cosas y tiempo, ocasionando que sea complicado para las pequeñas y medianas empresas abordar dichas estrategias. Entre los resultados que obtuvieron se demuestra que los procesos de Scrum pueden considerarse válidos bajo el paradigma CMMI, la evaluación proporcionan evidencias de que áreas de proceso relacionadas con el nivel 2 de CMMI-DEV fueron cubiertas en gran parte por Scrum.

Por otra parte, los autores (Akbar, Hassan, & Abdullah, 2012) en su trabajo: *A Framework of Software Process Tailoring for Small and Medium IT Companies* proponen el desarrollo de un framework para la adaptación de procesos de software basándose en los elementos críticos de las metodologías ágiles, esta propuesta surge por la problemática de que los trabajos existentes sobre la adaptación de procesos de software son muy generales y superficiales, además de no proveer guías para las pequeñas y medianas empresas de desarrollo de software. Entre los resultados obtenidos se comprueba que el framework identifica completamente elementos importantes de proyectos en pequeñas y medianas empresas, con los cuales se deriva el esquema para la adaptación del proceso.

Los autores (Al-Tarawneh et al., 2011) en su trabajo: *A Proposed Methodology for Establishing Software Process Development Improvement for Small Software Development Firms* proponen una metodología para establecer mejoras en los procesos de desarrollo de software utilizando CMMI-DEV 1.2 como modelo base para la mejora y el método ágil XP (eXtreme Programming) como método base para el desarrollo de software. La propuesta surge debido a la problemática de que los modelos de mejora de procesos de software están diseñados para empresas grandes y muy grandes, lo que hace difícil su implementación para pequeñas empresas, mismas que representan la mayoría de las empresas de software en la mayoría de los países. Entre los resultados obtenidos encontraron que con la metodología propuesta se les facilita a las PyMES la implementación de mejora de procesos con base en CMMI, a pesar de que este está diseñado para grandes empresas.

Por su parte los autores (Lin & Fan, 2009) en su trabajo: *Software Development Practice for FDA-Compliant Medical Devices* presentan una metodología híbrida que utiliza procesos ágiles para reducir el riesgo del proyecto, así como procesos formales basados en plan para la

creación de la documentación. La propuesta surge debido a la problemática de que el desarrollo de dispositivos médicos requiere una cantidad de documentación certera como evidencia de cumplimiento de procesos formales, lo cual se logra con el uso de modelos como CMMI, sin embargo, con el uso de métodos ágiles se lograría producir dispositivos con más alta calidad y mejor cumplimiento con las necesidades del mercado. Entre sus resultados obtuvieron que con esta metodología se cumplieron los requerimientos de la regulación FDA y al mismo tiempo se contó con un desarrollo ágil de software.

Los autores (Petersen & Wohlin, 2010) en su trabajo: *Software Process Improvement through the Lean Measurement (SPI-LEAM) method* presentan un método llamado SPI-LEAM donde, para evitar el riesgo de fallo, el método ayuda a la organización a alcanzar procesos Lean incrementalmente a través de mejora continua. La problemática que abordan los autores es que la manufactura Lean ha incrementado drásticamente calidad y la eficiencia en el desarrollo de productos, sin embargo, implementar Lean en desarrollo de software requiere un gran cambio de pensamiento sobre los procesos de software, por lo cual, un intento de cambiar toda la organización a la vez, a menudo conduce al fallo. Entre sus resultados encontraron que el método permite la mejora continua, la cual conduce a procesos de software Lean, además, evita problemas relacionados con la resistencia al cambio debido a que la mejora se desarrolla de manera continua.

Por otra parte, los autores (Vale et al., 2014) en su trabajo: *SPLICE: A Lightweight Software Product Line Development Process for Small and Medium Size Process* presentan a SPLICE: un proceso de desarrollo ligero que combina prácticas de Líneas de Productos de Software (SPLE) y prácticas ágiles. La propuesta surge debido a la problemática de que la combinación de líneas de productos de software y el desarrollo ágil de software es un enfoque que utilizan las empresas cuando trabajan en escenarios con requerimientos volátiles, en donde pueden hacer frente a los cambios rápidos, sin embargo, un proceso de desarrollo que cubra completamente el ciclo de vida SPLE utilizando prácticas ágiles en proyectos pequeños y medianos aún no se ha establecido. Los resultados de los casos de estudio muestran que se logran tres aspectos clave evaluados: Interacciones cortas y adecuadas, posibilidades para adaptar actividades y tareas, y retroalimentación continúa.

Los autores (Selleri Silva et al., 2015) en su trabajo: *Using CMMI together with agile: A systematic literature review* realizan un estudio para evaluar, sintetizar y presentar los resultados sobre el uso de CMMI en combinación con el desarrollo ágil de software, para después realizar un análisis de sus ventajas y limitaciones, así como las implicaciones para la investigación y la práctica. La problemática abordada por los autores es que la búsqueda de adhesión a los niveles de madurez mediante el uso de procesos ligeros que requieren bajos niveles de esfuerzo es considerado como un desafío para las organizaciones de desarrollo de software. Entre los resultados encontraron que las metodologías ágiles pueden ser utilizadas por organizaciones para reducir esfuerzos al obtener niveles 2 y 3 de madurez de CMMI, incluso se encontraron reportes de la aplicación de prácticas ágiles que condujeron a lograr el

nivel 5. Sin embargo, las metodologías ágiles por sí solas, de acuerdo a los estudios, no son suficientes para obtener un determinado nivel, siendo necesarias prácticas adicionales para lograrlo.

Por su parte los autores (Pino et al., 2010) en su trabajo: *Using Scrum to guide the execution of software process improvement in small organizations* proponen un proceso conocido como “Lightweight process to incorporate improvements”, el cual utiliza la filosofía del marco de desarrollo ágil Scrum y tiene el objetivo de dar guías detalladas para el apoyo a la gestión y la incorporación de oportunidades de mejora en los procesos, y su puesta en práctica en las pequeñas empresas. La problemática que abordan es que los modelos para mejora de procesos de software (SPI) no consideran las características especiales de las empresas pequeñas, ni las estrategias apropiadas para la implementación de SPI en estos tipos de organización, así mismo, los modelos que dirigen la aplicación de modelos en pequeñas organizaciones no presentan un proceso explícito para guiar el trabajo interno de los empleados involucrados en la ejecución de las oportunidades de mejora. Los resultados de la aplicación del proceso propuesto por medio de dos casos de estudio en dos pequeñas organizaciones muestran que el proceso es adecuado para este tipo de organizaciones:

- El esfuerzo necesario para administrar y llevar a cabo las tareas de mejora fue apropiado para los recursos de las dos pequeñas organizaciones.
- Con el uso de Scrum las oportunidades de mejora fueron incorporadas de una manera más cómoda que siguiendo un proceso rígido y prescriptivo. La mejora se dio de manera más “natural” para los empleados involucrados haciendo que las desapareciera cualquier resistencia inicial.

2.2.1. Comparativa de propuestas

A continuación en la Tabla 13 se muestra una comparativa de las propuestas mostradas anteriormente, incluyendo la problemática, la propuesta y los resultados.

Tabla 13. Comparativa de propuestas

ID	Problema	Propuesta	Resultados
P1	Es difícil para las PyMEs implementar estrategias de mejora con base a CMMI.	Comparación entre prácticas de CMMI nivel 2 y Scrum.	Se demuestra que los procesos de Scrum pueden considerarse válidos bajo el paradigma CMMI.
P2	Los trabajos existentes sobre la adaptación de procesos son muy generales y no proveen guías para las PyMEs.	Desarrollo de framework para adaptación de procesos, basado en los elementos críticos de las metodologías ágiles.	Un framework de adaptación de procesos ligero que genera un esquema de adaptación de procesos para PyMEs.
P3	Los modelos de mejora de procesos están diseñados para	Propuesta de metodología para establecer mejora de procesos	Se les facilita a las PyMES la implementación de mejora de

	grandes empresas y no para PyMES.	utilizando CMMI como modelo para la mejora y XP como método de desarrollo.	procesos con base en CMMI, a pesar de que este está diseñado para grandes empresas.
P4	El desarrollo de dispositivos médicos requiere procesos formales para cumplir con regulaciones, sin embargo, estos no entregan valor al cliente de manera directa.	Diseño de una metodología híbrida utilizando prácticas de CMMI y prácticas de procesos ágiles.	Se obtuvo un proceso ágil, mientras que se cumplieron los requerimientos de documentación para la regulación FDA.
P5	La implementación de Lean en desarrollo de software requiere un gran cambio, y tratar de cambiar toda la organización a la vez a menudo conduce al fallo.	Propuesta de SPI-LEAM (Software Process Improvement through the Lean Measurement), método que ayuda a las organizaciones a alcanzar procesos Lean de manera incremental a través de mejora continua.	El método permite el desarrollo de una mejora continua que conduce a procesos lean, a la vez que evita la resistencia al cambio.
P6	Aun no se ha establecido un proceso de desarrollo que cubra completamente el ciclo de vida de las líneas de productos de software utilizando prácticas ágiles.	SPLICE (Software Product Line Integrated Construction Environment): un proceso de desarrollo ligero que combina prácticas de líneas de productos de software y prácticas ágiles.	Se cumple con 3 aspectos clave evaluados: Interacciones cortas y adecuadas, posibilidades para adaptar actividades y tareas, y retroalimentación continua
P7	Un desafío para las organizaciones es la adhesión a niveles de madurez mediante el uso de procesos ligeros que requieren bajos niveles de esfuerzo.	Se aplica el método de revisión sistemática para evaluar, sintetizar y presentar los resultados sobre el uso de CMMI en combinación con desarrollo ágil de software.	Se pueden alcanzar los niveles 2 y 3 de CMMI con el uso de prácticas ágiles, sin embargo, por sí solas, las metodologías ágiles no son suficientes para obtener un nivel, requiriendo prácticas adicionales.
P8	Los modelos de SPI no toman en cuenta las características de las PyMEs, ni presentan un proceso explícito para guiar la ejecución de la mejora.	Propuesta de un proceso ligero para incorporar mejoras en pequeñas empresas utilizando al método Scrum para guiar la implementación de mejoras.	El esfuerzo necesario para las actividades de mejora fue apropiado para los recursos de la organización y gracias a Scrum las oportunidades de mejora se incorporaron de manera más cómoda.

2.2.2. Características cubiertas por los trabajos relacionados

A continuación en la Tabla 14 se analizan el grado en el que las estrategias identificadas para aligerar procesos son cubiertas por las propuestas analizadas, así mismo se agrega el modelo, estándar o metodología ágil que implementa. Para el análisis se utiliza la siguiente nomenclatura: si una característica es cubierta completamente; se utiliza el icono relleno con

color negro (●), si la característica es cubierta parcialmente; se utiliza el icono relleno a la mitad (◐), por último, si la característica no es cubierta; se utiliza el icono blanco (○).

Tabla 14. Características cubiertas por los trabajos relacionados

Característica		2.2.1	2.2.2	2.2.3	2.2.4	2.2.5	2.2.6	2.2.7	2.2.8
Combina prácticas formales y prácticas ágiles		●	◐	●	●	◐	●	●	◐
Utiliza principios Lean		○	○	○	○	●	○	○	○
Identifica las mejores prácticas de la organización		○	◐	○	◐	◐	○	○	◐
Identifica factores de éxito de las metodologías ágiles		○	●	○	○	○	◐	◐	○
Modelo / Estándar base:	CMMI	●	○	●	●	○	○	●	○
	ISO/IEC 15504	○	○	○	○	○	○	○	○
Método ágil:	Scrum	●	○	○	◐	○	●	◐	●
	XP	○	○	●	◐	○	○	◐	○

Una vez analizados los trabajos relacionados, se concluye que no hay ningún trabajo relacionado cuya propuesta cubra completamente más de una estrategia para aligerar procesos. Por lo cual, como parte de esta investigación se propone un método para aligerar procesos de software utilizando las siguientes estrategias: combinación de prácticas formales y prácticas ágiles, uso de principios Lean e identificación de las mejores prácticas de la organización. La propuesta del método se describe a detalle en el Capítulo 4.

2.3. Selección de tecnología

Junto a la propuesta del método para aligerar procesos, se va a desarrollar una herramienta que facilite su uso, la cual es conveniente que se desarrolle como una aplicación Web, de modo que pueda ser accesible desde cualquier plataforma por medio de un navegador Web, y sin necesidad de realizar instalación de software adicional. Por lo cual es necesario seleccionar tecnología (de preferencia software libre para evitar pagos de licencias) que permita el desarrollo de la herramienta como aplicación Web.

2.3.1. Frameworks de desarrollo Web

Hay una amplia variedad de lenguajes de programación enfocados en el desarrollo de aplicaciones Web, entre los que destacan: PHP, Python y Ruby, a su vez, para estos lenguajes existen distintos Frameworks Web que aceleran el proceso de desarrollo, ya que reutilizan

código existente y promueven buenas prácticas de desarrollo como el uso de patrones. Para los lenguajes de programación mencionados anteriormente los frameworks de código abierto más populares son: Laravel (PHP), Django (Python) y Ralis (Ruby). A continuación se muestran algunas de sus características principales.

2.3.1.1. **Laravel**

Laravel es un framework Web de código abierto escrito en PHP, creado por Taylor Otwell y enfocado en el desarrollo de aplicaciones Web siguiendo el patrón arquitectónico Modelo-Vista-Controlador (Otwell, 2016).

Entre sus características principales se encuentran

- Sintaxis expresiva: Simple y legible.
- Sistema de rutas: Flexibilidad y control sobre el direccionamiento.
- Pruebas unitarias: Permite ejecutar cientos de pruebas para asegurar que los nuevos cambios no afectan el funcionamiento de la aplicación.
- Eloquent ORM (Mapeo Objeto-Relacional): es un avanzado patrón que proporciona métodos internos para relacionar los objetos con tablas de la base de datos.
- Paginación automática: Simplifica la tarea de implementar paginación con métodos automatizados.

2.3.1.2. **Django**

Django es un framework Web de Python de alto nivel que fomenta el rápido desarrollo y diseño limpio y pragmático. Construido por desarrolladores con experiencia, se encarga de gran parte de las actividades comunes en el desarrollo Web, para que el desarrollador pueda centrarse en la escritura de su aplicación sin necesidad de reinventar la rueda. Es de código abierto y libre (Django Software Foundation, 2016).

Entre sus características principales se encuentran:

- Utiliza el patrón: MTV: Modelo-Plantilla-Vista
- Desarrollo rápido: Django fue diseñado para llevar una aplicación desde el concepto hasta su finalización lo más rápido posible.
- Seguridad: Ayuda a los desarrolladores a evitar errores comunes de seguridad como: inyección SQL, cross-site scripting, cross-site request forgery y clickjacking.
- Escalable: Permite escalar con rapidez y flexibilidad para satisfacer demandas de tráfico pesado.

- Versátil: Organizaciones, empresas y gobierno han usado Django para construir una gran variedad de cosas como: sistemas de administración de contenido, redes sociales y plataformas de cómputo científico.

2.3.1.3. **Ruby on Rails**

Es un entorno de desarrollo Web de código abierto que está optimizado para la satisfacción de los programadores y para la productividad sostenible. Fue creado en 2003 por David Heinemeier Hansson y desde entonces ha sido extendido por una comunidad de más de 4,200 personas (Hansson, 2016).

Algunas de sus características principales son:

- Convención sobre configuración.
- Promueve buenas prácticas de desarrollo: Mantenlo simple (keep it simple), No te repitas (Don't repeat yourself), entre otras.
- Utiliza el patrón: Modelo-Vista-Controlador.
- Expresividad en el código.
- Framework seguro: incluye características para proteger a la aplicación de ataques como: CRSF, Script Injection, SQL Injection, entre otros.
- Resuelve los problemas comunes al desarrollar una aplicación Web, para que el programador pueda concentrarse en los aspectos únicos y diferenciales de sus proyectos.

2.3.1.4. **Comparativa de Frameworks**

Una vez que se han analizado algunas de las principales características de estos tres frameworks, se realiza una comparativa entre las características necesarias para el desarrollo de la herramienta.

Como se puede observar en la

Tabla 15, los frameworks cumplen con casi todas las características, sin embargo se opta por Ruby on Rails como framework de desarrollo por ser el que tiene la comunidad de soporte más extensa.

Tabla 15. Características necesarias para el desarrollo de la herramienta

Característica	Laravel	Django	Ruby on Rails
Software libre y de código abierto.	✓	✓	✓
Herramientas de seguridad	✓	✓	✓
Comunidad de apoyo extensa	-	-	✓
Patrón de diseño	MVC	MTV	MVC
Creación automática de tablas en la base de datos a partir de los modelos	✓	✓	✓
Sistema de direccionamiento fácil y seguro	✓	✓	✓
Desarrollo rápido	✓	✓	✓
Promueve buenas prácticas de desarrollo	-	✓	✓

2.3.2. Gestores de Bases de datos

Además de un framework de desarrollo Web es necesario contar con un sistema gestor de base de datos (DBMS) que permita hacer un CRUD (Crear, Leer, Actualizar y Borrar) de datos para administrar la información que maneja la aplicación Web.

Entre los DBMS más populares y que son software libre se encuentran: MySQL, PostgreSQL y MariaDB. A continuación se muestran sus características principales.

2.3.2.1. MySQL

Es la base de datos más popular del mundo, se ha convertido en la opción principal de bases de datos para las aplicaciones basadas en Web. Actualmente es desarrollada por Oracle Corporation, y cuenta con una versión libre con licencia GNU GPL y otras versiones comerciales para empresas (Oracle Corporation, 2016).

Entre sus características principales se encuentran:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferentes velocidades de operación, soporte físico, capacidad o distribución geográfica.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

2.3.2.2. **MariaDB**

MariaDB es uno de los servidores de bases de datos más populares en el mundo. Construida por los desarrolladores originales de MySQL y garantizado para permanecer como software libre. MariaDB es una bifurcación directa de MySQL por lo cual hay una alta compatibilidad entre ambos (MariaDB Foundation, 2016).

Entre sus características principales se encuentran:

- Rápida, escalable y robusta.
- Gran cantidad de motores de almacenamiento (más que MySQL).
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Transacciones y claves foráneas.
- Replicación.
- Búsqueda e indexación de campos de texto.

2.3.2.3. **PostgreSQL**

PostgreSQL es un potente sistema de base de datos objeto-relacional, de código abierto. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación de fiabilidad, integridad de datos y la corrección. A diferencia de otros proyectos, PostgreSQL no es manejado por una empresa, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada (The PostgreSQL Global Development Group, 2016).

Entre sus características principales se encuentran:

- Altamente personalizable.
- Alta concurrencia.
- Transacciones y claves foráneas.
- Disparadores (triggers).
- Complimiento con estándares.
- Diseñado para entornos de alto volumen.
- Multiplataforma: compatible con 34 plataformas incluyendo cualquier marca de Unix.

2.3.2.4. Comparativa de gestores de bases de datos

Como se puede observar en la Tabla 16, en términos generales, las bases de datos de código abierto analizadas comparten entre sí sus características principales.

Tabla 16. Comparativa de gestores de bases de datos

Modelo	MySQL	MariaDB	PostgreSQL
Sistema gestor de bases de datos relacional	✓	✓	✓
ACID: Atomicidad, Consistencia, Aislamiento, Durabilidad.	✓	✓	✓
Tipo de licencia	GPL	GPL v2	PostgreSQL License
Multiplataforma	✓	✓	✓
Transacciones y respaldos	✓	✓	✓
Claves foráneas	✓	✓	✓
Replicación	✓	✓	✓

Para el desarrollo de la herramienta se elige MariaDB debido a que además de estar basado y ser compatible con el gestor MySQL, su desarrollo es más abierto, y cuenta con más motores de almacenamiento que MySQL, lo cual lo hace que tenga un mejor rendimiento.

2.4. Metodología de desarrollo

Existe una gran cantidad de metodologías enfocadas en el desarrollo de aplicaciones Web, entre las más populares se encuentran: UWE, WebML y OOHDM.

2.4.1. UWE

UWE (UML Web-based) es una metodología que tiene como objetivo cubrir el ciclo de vida del desarrollo de aplicaciones web. El aspecto clave que distingue a UWE es la confianza en estándares (Web Engineering Group, 2015) .

El enfoque principal de UWE es proporcionar:

- Lenguaje de modelado de dominio específico basado en UML.
- Una metodología basada en modelos.
- Herramientas de soporte para el diseño sistemático.
- Herramientas de soporte para la generación (semi-)automática de aplicaciones Web.

Consta de 6 modelos:

- Modelo de requerimientos. Consiste definir los casos de uso de la aplicación y sus relaciones.
- Modelo de contenido. Equivale a un diagrama de clases normal del UML.
- Modelo de navegación. Este modelo permite conocer la manera en la que las páginas web está enlazadas, se compone solamente de nodos y enlaces,
- Modelo de presentación. Permite conocer a que parte de la página web pertenecen las clases de navegación y procesos.
- Modelo de proceso. Permite conocer las acciones de las clases, tiene dos componentes: la estructura del proceso y el flujo del proceso.

2.4.2. **WebML**

WebML (Web Modeling Language) es una notación visual y metodológica para el diseño de aplicaciones Web complejas que usan datos intensivamente (Politecnico di Milano, 2003).

Los principales objetivos de la metodología son:

- Expresar la estructura de una aplicación web con una descripción de alto nivel.
- Proporcionar múltiples vistas a un mismo contenido.
- Separar la información de su composición en páginas, navegación y presentación.
- Almacenamiento de meta-información recogida durante el proceso de diseño, para la generación dinámica de páginas web.
- Permitir la especificación de las operaciones de manipulación de datos para actualizar el contenido del sitio o interactuar con los servicios externos arbitrarios.

Los modelos propuestos por WebML para el diseño de páginas web son:

- Modelo de datos
- Modelo de hipertexto (Composición y navegación del sitio)
- Modelo de presentación
- Modelo de personalización

2.4.3. OOHDM

OOHDM (Object Oriented Hypermedia Design Method) es un método para el diseño de hipertexto y aplicaciones Web, fue diseñado por Daniel Schwabe y Gustavo Rossi (Daniel Schwabe, 2008).

Las piedras angulares del enfoque OOHDM son:

- La idea de que los objetos de navegación son vistas.
- El uso apropiado de abstracciones para organizar el espacio de navegación.
- La separación de los problemas de interfaz de asuntos de la navegación.
- Una identificación explícita sobre que hay decisiones de diseño que tendrán que ser establecidas en el momento de ejecución.

Los modelos propuestos en OOHDM son:

- Modelo conceptual
- Diseño de navegación
- Diseño de interfaces abstractas

2.4.4. Comparación de metodologías

Las tres metodologías analizadas anteriormente comparten modelos en común, en la Tabla 17 se observa que los modelos de datos, navegación y presentación son cubiertos por las tres metodologías.

Tabla 17. Modelos propuestos en las metodologías

Modelo	UWE	Webml	OOHDM
Modelo de casos de uso	✓		
Modelo de datos/contenido	✓	✓	✓
Modelo de navegación	✓	✓	✓
Modelo de presentación	✓	✓	✓
Modelo de personalización		✓	
Modelo de proceso	✓		

Para el desarrollo de la herramienta se tomará como base la metodología UWE, ya que ésta no sólo se enfoca en la fase de diseño, sino que cubre todo el ciclo de vida del desarrollo

de aplicaciones web; desde el modelado de requerimientos mediante la elaboración de casos de uso hasta el modelo de la presentación.

Capítulo 3. Metodología para el desarrollo de la tesis

En este capítulo se describe la metodología utilizada para el desarrollo de la tesis. La Figura 8 muestra las tareas llevadas a cabo durante el desarrollo de la tesis y el orden en que se implementaron.

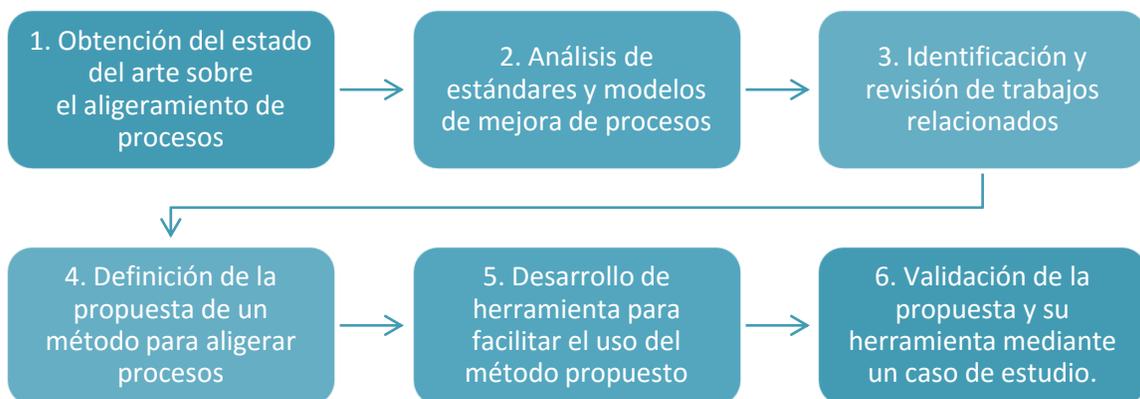


Figura 8. Metodología para el desarrollo de la tesis.

Como se observan en la Figura 8 se inició con la obtención del estado del arte sobre el aligeramiento de procesos, el cual fue obtenido mediante el desarrollo de una revisión sistemática. La siguiente tarea consistió en identificar y analizar estrategias para el aligeramiento de procesos, las cuales fueron identificadas como parte de los principales resultados de la revisión sistemática. Posteriormente se realizó un análisis de los principales estándares y modelos de mejora de procesos de software. En seguida se identificaron un conjunto de trabajos relacionados para revisar las características que cumplían. A continuación se definió la propuesta de un método para aligerar procesos a partir de las estrategias identificadas anteriormente. Consecutivamente a la definición de la propuesta se diseñó y desarrolló una herramienta para facilitar el uso del método. Finalmente, se realiza una validación para comprobar la viabilidad del método propuesto.

3.1. Ejecución de la metodología para el desarrollo de Tesis

- 1. Establecimiento del estado del arte sobre el aligeramiento de procesos.** Con el objetivo de obtener un panorama actual sobre el aligeramiento de procesos se realiza una revisión sistemática de la literatura enfocada en tres aspectos principales: 1) Frameworks, métodos y metodologías, 2) Procesos enfocados y 3) estrategias para aligerar procesos. En el Capítulo 2, sección 2.1 se encuentra detallado el protocolo de la revisión sistemática desarrollados y los resultados obtenidos.
- 2. Análisis de modelos y estándares de mejora de procesos.** Se realizó un análisis de modelos y estándares de mejora de procesos, incluyendo a CMMI, ISO/IEC 15504, MoProSoft e ISO/IEC 29110. Para cada uno de los modelos y estándares se escribe su descripción, niveles de madurez y, áreas de procesos. El análisis completo se encuentra dentro del marco teórico del Capítulo 1, en la sección 1.1.3.
- 3. Identificación y revisión de trabajos relacionados.** A partir de la revisión sistemática se eligieron un conjunto de trabajos que muestran propuestas para aligerar procesos de software y que utilizan las estrategias identificadas anteriormente (excluyendo el uso de herramientas). Se realiza un análisis de dicho trabajos mediante una comparativa de las propuestas, y así mismo se examinan las características cubiertas por los trabajos. El desarrollo de esta actividad se encuentra en el capítulo 2, sección 2.2.
- 4. Definición de la propuesta de un método para aligerar procesos.** Con base a las siguientes estrategias: uso de principios Lean, identificación de las mejores prácticas de la organización y combinación de prácticas formales con prácticas ágiles, se realiza la propuesta de un método para aligerar procesos el cual se compone de los siguientes pasos: modelado del proceso de la empresa, análisis del valor añadido, mapeo de prácticas formales a prácticas ágiles, sugerencia de técnicas y herramientas y, rediseño del proceso. La definición y descripción de los pasos del método se encuentra en el Capítulo 4, secciones 4.1 y 4.2.
- 5. Desarrollo de la herramienta para facilitar el uso del método propuesto.** A partir de la propuesta del método para aligerar procesos se desea y desarrolla una herramienta que facilite su uso, la tecnología utilizada se describe en la sección 2.3, mientras que la metodología de desarrollo se encuentran en la sección 2.4, ambas secciones se encuentran dentro del Capítulo 2. El desarrollo de la herramienta se detalla en el Capítulo 4, sección 4.4.
- 6. Validación de la propuesta en conjunto con su herramienta.** Para validar la herramienta y la propuesta del método se realizaron dos casos de estudio en empresas

de desarrollo de software con características diferentes. El detalle de los casos de estudio se encuentran en el Capítulo 5, sección 5.2.

Capítulo 4. Método para aligerar procesos

Este capítulo contiene la propuesta del método para aligerar procesos de software. Se comienza con la especificación de las estrategias utilizadas para diseñar el método. En seguida se muestran los requisitos que deben cumplirse antes de poder aplicar el método. Posteriormente se detallan a profundidad cada uno de los pasos que componen al método. Por último, se muestra el desarrollo de la herramienta que facilita el uso del método.

4.1. Desarrollo del método

A partir de las estrategias para aligerar procesos identificadas en la revisión sistemática, se seleccionaron las siguientes estrategias como base para la propuesta del método:

1. Identificación de las mejores prácticas de la organización
2. Uso de principios Lean.
3. Combinación de prácticas formales y prácticas ágiles.

A pesar de que la estrategia más utilizada es el uso de herramientas, esta no fue seleccionada como estrategia, debido a que las herramientas se enfocan en la automatización del proceso y no en análisis del proceso o de las actividades del proceso para su optimización. Sin embargo, en la propuesta se tiene contemplado hacer sugerencias de herramientas y técnicas para llevar a cabo la implementación de las prácticas.

El método está dirigido a empresas de desarrollo de software con experiencia en la implementación de mejora de procesos o empresas sin experiencia pero con inquietud de la implementación de mejoras.

La propuesta del método se compone de 5 pasos (Figura 9): se comienza con el modelado del proceso de la empresa, se analiza el valor añadido que tienen las prácticas para remover las que no aporten valor, se mapea las prácticas formales con prácticas ágiles, se realizan

sugerencias de técnicas y herramientas y, por último, se rediseña el proceso con los cambios sugeridos. Como el método es “mejora continua”, puede ejecutarse de manera iterativa.



Figura 9. Propuesta del método para aligerar procesos

Cada uno de los pasos del método se detalla a mayor profundidad en la sección 4.3.

4.2. Prerrequisitos para el uso del método

La implementación del método debe cumplir con los siguientes prerrequisitos para el buen funcionamiento del método:

- 1) La empresa debe ser una empresa dedicada al desarrollo de software con o sin experiencia en la implementación mejora de procesos.
- 2) Identificar un encargado de la implementación de mejoras de procesos, idealmente perteneciente al Grupo de Procesos de Ingeniería de Software (SEPG, por sus siglas en inglés: Software Engineering Process Group).
- 3) Calcular una muestra de empleados significativa basada en el número total de empleados de la empresa. Se sugiere utilizar la siguiente fórmula para el cálculo del tamaño de la muestra con los siguientes datos:

$$n = \frac{N\sigma^2 Z_a^2}{e^2(N-1) + \sigma^2 Z_a^2}$$

Donde:

N = Tamaño de la empresa

σ = Desviación estándar, cuando no se tiene su valor se utiliza 0.5, debido a que los empleados de la empresa se encuentran en el mismo entorno la variabilidad de su distribución debe ser menor, por lo que se recomienda utilizar 0.3.

Z_{α} = Nivel de confianza, se propone utilizar 1.28 para dar un nivel de confianza del 80%.

e = Límite aceptable de error, suele utilizar se un valor entre 1% (0.01) y 10% (0.1).

Ejemplo: Para una empresa con un tamaño de 50 el tamaño de muestra es:

$$n = \frac{50(0.3)(1.28)^2}{(0.1)^2(50 - 1) + 0.3(1.28)^2} = 12$$

4.3. Descripción de los pasos del método

Para la descripción de los pasos del método se toman en cuenta los siguientes elementos: propósito, información de entrada, fases/actividades, roles y productos de trabajo, los cuales se describen a continuación:

- Propósito: sentencia clara y concisa que describe los objetivos y razón de la fase.
- Información de entrada: es el recurso que es transformado por la fase en una salida (producto de trabajo) o la información necesaria para ejecutar las actividades de la fase.
- Fases/actividades: etapas o pasos intermedios de un proceso y el orden en que se realizan.
- Roles: quién y que función desempeña en la realización de cada una de las actividades de la fase.
- Productos de trabajo: son los productos generados por un paso del método, pueden ser: a) salidas, son productos o servicios que resultan de una fase, y b) salidas intermedia, se convierten entradas de otras fases del proceso.

4.3.1. Paso 1: Modelado del proceso de la empresa

El primer paso del método es el modelado del proceso de la empresa, para lo cual es necesario extraer las prácticas que actualmente implementa la organización y posteriormente modelar el proceso.

4.3.1.1. **Propósito**

Extraer las prácticas que realmente son llevadas a cabo en la organización y diagramar un proceso en base a ellas.

4.3.1.2. **Información de entrada**

- Cuestionario para la obtención de prácticas tomando CMMI como modelo de referencia para formalizar las prácticas.
- Reunión inicial para explicación del método y selección del encargado de mejores y usuarios del proceso.

4.3.1.3. **Fases/actividades**

El modelado del proceso de la empresa consta de dos actividades: aplicación de cuestionarios y diagramado del proceso, las cuales se detallan a continuación.

A1. Aplicación de cuestionarios

Para la extracción de las prácticas implementadas en la organización se realiza un cuestionario con preguntas sobre las prácticas específicas del área de proceso de planificación de proyectos del modelo CMMI. Cada pregunta incluye ejemplos que le permitan a la persona que contesta el cuestionario saber si se implementa o no dicha práctica en la organización. Así mismo, las preguntas son cerradas y entre las respuestas solo se puede elegir: sí, no y no sabe.

Ejemplo de preguntas:

1. ¿Se estima el alcance del proyecto?

Ejemplo de estimaciones de alcance:

- a. Establecer un WBS (Estructura de descomposición de trabajo) para estimar el alcance del proyecto.

- a) Sí b) No c) No sabe

El cuestionario completo para el área de proceso planificación de proyecto se encuentra en el Anexo C.

A2.Diagramado del proceso

El diagramado del proceso consiste en tomar las prácticas implementadas en la organización y modelar el proceso de acuerdo a la especificación BPMN (Modelo y Notación de Procesos de Negocio).

En esta fase sólo participa un representante del SEPG (Grupo de Procesos de Ingeniería de Software), el cual se encargará de diagramar el proceso con ayuda de la herramienta bpmn modeler de bpmn.io. El SEPG contará con un grupo de elementos BPMN (Actividades) que contienen las prácticas implementadas en la organización y, su tarea es establecer conexiones, eventos, compuertas u otros elementos BPMN para representar el proceso actual de la organización.

4.3.1.4. Roles

Para esta fase participan dos roles:

- SEPG: encargado de las actividades de mejora de procesos de software en la organización.
- Empleado de la empresa: trabajador de la organización que usa los procesos de software de la organización.

La Tabla 18 muestra las actividades en las que participan los usuarios mencionados.

Tabla 18. Roles para el paso 1: Modelado del proceso de la empresa

Actividad	Roles	
	SEPG	Empleado
Aplicación de cuestionarios	✓	
Diagramado del proceso	✓	✓

4.3.1.5. Productos de trabajo

- Lista de prácticas específicas que se llevan a cabo en la organización.
- Diagrama del proceso de la organización “As is” (tal como está).

4.3.2. Paso 2: Análisis del valor añadido

El análisis del valor añadido es uno de los principios Lean más importantes, el cual se enfoca en la eliminación de desperdicios dentro de los procesos que no contribuyen a la creación de valor para el cliente o la empresa (Petersen & Wohlin, 2010).

Este paso consiste en analizar el valor que aportan las prácticas de la empresa de acuerdo a una muestra de empleados determinada como prerrequisito del método, con el objetivo de quitar las prácticas que no estén aportando valor.

Con el análisis del valor añadido, además de identificar las actividades que no aportan valor a la empresa, también se identifican cuáles son las mejores prácticas de la organización de acuerdo a los empleados de la empresa.

4.3.2.1. Propósito

Identificar las prácticas del proceso que no contribuyen a la creación de valor para el cliente o la empresa, así como obtener las mejores prácticas de la organización.

4.3.2.2. Información de entrada

- Lista de prácticas específicas implementadas en la organización.

4.3.2.3. Fases/actividades

Este paso consta de dos actividades: capturar el valor añadido de las prácticas y analizar los valores añadidos para encontrar las prácticas que no aportan valor.

A1. Captura el valor añadido

La captura del valor añadido se realiza mediante una escala de valoración. La escala de valoración más común en investigación es la escala Likert, la cual contiene ítems con opciones que van desde: totalmente en desacuerdo hasta totalmente de acuerdo. Sin embargo, para capturar el valor de una práctica es necesario una escala donde la valoración sea numérica, para posteriormente poder aplicar cálculos que conduzcan a las prácticas que no aportan valor.

Por esta razón la captura del valor añadido se realiza mediante en una escala numérica que va de 0 a 5. En aplicaciones web uno de los patrones UX (Experiencia de Usuario) más conocidos es el “star rating” (calificación con estrellas), utilizado por aplicaciones web de grandes compañías como Facebook, Amazon, Netflix, entre otros.

En esta actividad la tarea del usuario (SEPG o empleado) es asignar un valor o calificación por medio de estrellas a cada una de las prácticas implementadas en la organización.

A2. Análisis del valor añadido

Para cada una de las prácticas se realiza un análisis de su valor, el cual se calcula mediante los siguientes pasos:

1. Calcular el promedio a partir de los valores asignados para cada práctica:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

Siendo:

n : número de veces que se calificó una práctica

x_i : el valor asignado a cada práctica

\bar{x} : promedio de valores asignados a la práctica

2. Eliminar outliers (valores atípicos) y calcular nuevamente el promedio. Se consideran como outliers a los valores de las prácticas que tengan una diferencia mayor o igual a 3 con respecto al promedio, y a la vez la cantidad de dichos valores sea menor al 25% del número total de calificaciones.
3. Obtener el número índice del promedio en base a 100:

$$I = \frac{\bar{x}}{5} * 100$$

Siendo:

\bar{x} : promedio de valores

I = número índice

Agrupar el número índice en alguno de los rangos mostrados en la Tabla 19 y ejecutar la acción indicada en la

4. Figura 10

Tabla 19. Rangos para el análisis del valor añadido

Rango	Análisis del valor añadido
0 – 25	No aporta valor
26 – 50	Aporta poco valor
51 – 75	Aporta valor
76 - 100	“Mejor práctica”

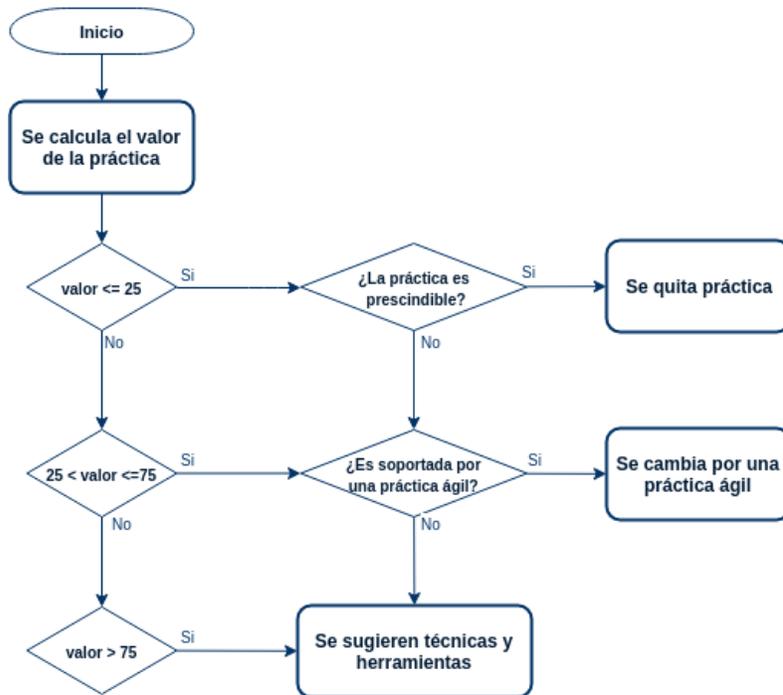


Figura 10. Acción a ejecutar de acuerdo al valor de la práctica.

4.3.2.4. Roles

A continuación en la Tabla 20 se muestran los roles que participan en este paso, así como sus correspondientes actividades.

Tabla 20. Roles para el paso 2: Análisis del valor añadido

Actividad	Roles	
	SEPG	Empleado
Captura del valor añadido	✓	✓
Análisis del valor añadido		

4.3.2.5. Productos de trabajo

- Lista de prácticas específicas con su respectivo valor asignado.
- Lista de prácticas que no aportan valor y son candidatas a eliminarse.

4.3.3. Paso 3: Mapeo de prácticas formales a prácticas ágiles

Una de las estrategias (encontradas en la revisión sistemática) más utilizadas para aligerar procesos de software es la combinación de prácticas formales y prácticas ágiles. Esta estrategia se incluye en el método mediante la realización de un mapeo de prácticas formales con sus equivalentes en prácticas ágiles.

Para el mapeo de prácticas se toman como base los estudios realizado por (Marçal, De Freitas, Furtado Soares, & Belchior, 2007) y por (Diaz et al., 2009), en los cuales se realiza un mapeo entre las prácticas específicas pertenecientes a las áreas de procesos de la categoría de Gestión de Proyectos de CMMI a prácticas de Scrum.

En el estudio realizado por (Marçal et al., 2007) para el mapeo de prácticas se estableció una calificación de cobertura para cada práctica teniendo en cuenta los criterios: no satisfecha, parcialmente satisfecha y completamente satisfecha. Entre los resultados de este estudio se encontró que los procesos de la categoría de Gestión de Proyectos de CMMI sólo son cubiertos en un 31.1% por Scrum. Así mismo, las áreas de proceso más cubiertas por prácticas de Scrum son PP con un 57.1% y PMC con un 50% (ver Figura 11).

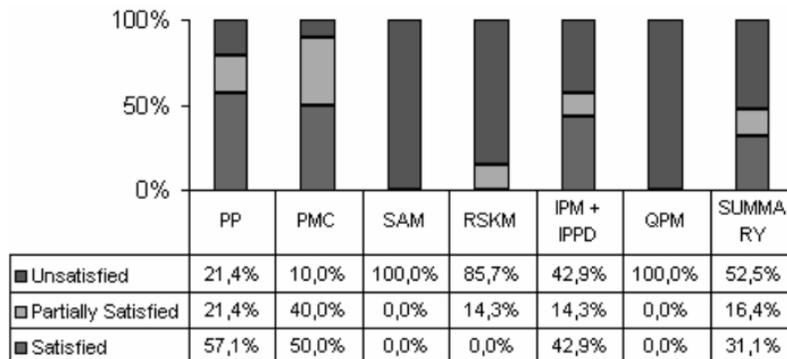


Figura 11. Áreas de Proceso de la categoría de Gestión de Proyectos de CMMI cubiertas por Scrum (Marçal et al., 2007).

Para el desarrollo de este método, como prácticas formales se toman las prácticas específicas del área de proceso de Planificación del Proyecto debido a que es el área de proceso más cubiertas por Scrum. Así mismo, Scrum es el marco de desarrollo ágil más utilizado en la industria de software (Mundra, Misra, & Dhawale, 2013).

4.3.3.1. Propósito

Identificar prácticas de Scrum que cubran completa o parcialmente a las prácticas que se llevan a cabo en la organización.

4.3.3.2. Información de entrada

- Lista de prácticas específicas que se llevan a cabo en la organización.

4.3.3.3. Fases/actividades

Este paso consta sólo de una actividad: el mapeo de prácticas específicas a prácticas de Scrum, el cual se detalla a continuación.

A1. Mapeo de prácticas específicas a prácticas de Scrum

Para el mapeo de prácticas específicas a Scrum se realiza una tabla de equivalencia de prácticas específicas del área de proceso planificación de proyectos a prácticas de Scrum tomando como base los trabajos de (Marçal et al., 2007) y (Díaz et al., 2009). Esta tabla se tomará como base para la aplicación del método. A continuación en la Tabla 21 se muestra un ejemplo del mapeo entre prácticas: para cada práctica específica se muestra una práctica de Scrum equivalente. Además, cada práctica cuenta con una descripción sobre cómo se puede llevar a cabo la práctica ágil.

Tabla 21. Mapeo de prácticas específicas a prácticas ágiles

Estado	Practica específica	Práctica de Scrum
●	<p>1.1. Estimar el alcance del proyecto</p> <p><i>Ejemplo:</i> Establecer un WBS (Estructura de descomposición de trabajo) para estimar el alcance del proyecto.</p>	<p>Establecer WBS a partir del Product Backlog</p> <p><i>Descripción:</i> En la fase pre-game los stakeholders pueden contribuir a la creación del Product Backlog (Pila de producto). En este caso el WBS se compone del Product Backlog y el conjunto de los sprints predefinidos, proporcionando los recursos necesarios para estimar el alcance del proyecto. Los estimados detallados se llevan a cabo al inicio de cada sprint.</p>
●	<p>1.2. Establecer las estimaciones de los atributos de los productos de trabajo y de las tareas.</p> <p><i>Ejemplos de atributos para estimar son:</i></p> <ul style="list-style-type: none"> • Número y complejidad de los requisitos. • Número y complejidad de las interfaces. • Número de funciones. • Puntos de función. 	<p>Estimar atributos mediante Planning Poker</p> <p><i>Descripción:</i> Scrum establece una primera estimación en la fase pre-game y una estimación iterativa al principio del sprint (reunión de planificación). Las estimaciones se basan generalmente en tamaño o complejidad atributos. Algunas prácticas ágiles recomiendan la técnica de estimación Planning Poker; que se basa en el consenso de los participantes para estimar el tamaño relativo de las historias de usuario. Algunas unidades pueden incluir puntos de la historia o puntos de función.</p>

La tabla completa de mapeo de prácticas específicas en el área de Planificación de Proyectos a prácticas de Scrum se encuentra en el Anexo D.

4.3.3.4. **Roles**

Este paso no tiene roles, debido a que el mapeo de prácticas se ha definido con el desarrollo del método y su aplicación.

4.3.3.5. **Productos de trabajo**

- Lista de prácticas específicas del área de proceso PP con una relación a prácticas equivalentes en Scrum

4.3.4. **Paso 4: Sugerencia de técnicas y herramientas**

Este paso consiste en proporcionar sugerencias de técnicas y herramientas para la implementación de las prácticas, de modo que las técnicas en conjunto con las herramientas ayuden a la organización a implementar sus prácticas de una manera más ágil y eficiente.

4.3.4.1. **Propósito**

Proporcionar sugerencias de técnicas y herramientas para facilitar la implementación de las prácticas de la organización.

4.3.4.2. **Información de entrada**

- Lista de prácticas específicas implementadas en la organización.

4.3.4.3. **Fases/actividades**

Este paso sólo consta de una actividad: que se enfoca en sugerir un conjunto de técnicas y herramientas ligeras.

A1.Sugerencia de técnicas y herramientas

Para hacer las sugerencias de técnicas y herramientas ligeras se implementa una tabla en la cual cada práctica específica del área de planificación de proyectos contiene un conjunto de técnicas y herramientas asociadas.

Un ejemplo de la tabla que contiene las técnicas y herramientas asociadas a cada práctica se encuentra en la

Tabla 22.

Tabla 22. Sugerencia de técnicas y herramientas

Practica específica	Técnicas o herramientas
<p>1.1. Estimar el alcance del proyecto <i>Ejemplo:</i> Establecer un WBS (Estructura de descomposición de trabajo) para estimar el alcance del proyecto.</p>	<ul style="list-style-type: none"> • Plantillas WBS de proyectos previos • Información Histórica • Descomposición que comprende la subdivisión en entregables y sub-entregables
<p>1.2. Establecer las estimaciones de los atributos de los productos de trabajo y de las tareas <i>Ejemplos de atributos para estimar son:</i></p> <ul style="list-style-type: none"> • Número y complejidad de los requisitos. • Número y complejidad de las interfaces. • Número de funciones. • Puntos de función. 	<ul style="list-style-type: none"> • Información Histórica • Juicio de especialistas • Información Histórica • Líneas de código o puntos función para el software • Número/ complejidad de requerimientos para la ingeniería de sistemas • Estimaciones Top-Down • Estimaciones Bottom-Up • Plantillas en donde se describan las actividades, recursos de habilidad, tiempo y esfuerzo requerido, identificación de riesgos

4.3.4.4. Roles

Este paso no tiene roles, debido a que la sugerencias de técnicas y herramientas no involucra a los usuarios del método.

4.3.4.5. Productos de trabajo

- Lista de prácticas específicas del área de PP con técnicas y herramientas sugeridas para cada práctica.

4.3.5. Paso 5: Rediseño del proceso

El último paso del método consiste en rediseñar el proceso establecido en el paso 1, de manera que se puedan visualizar los cambios y sugerencias aplicadas en los pasos posteriores.

4.3.5.1. Propósito

Rediseñar el proceso de la organización con los cambios recomendados: quitando prácticas que no añaden valor, cambiando prácticas formales por prácticas ágiles y realizando sugerencias de técnicas y herramientas ligeras.

4.3.5.2. **Información de entrada**

- Lista de prácticas específicas que se llevan a cabo en la organización.
- Diagrama del proceso de la organización “As is” (tal como está).
- Lista de prácticas que no aportan valor y son candidatas a eliminarse.
- Lista de prácticas implementadas en la organización con una relación a prácticas equivalentes en Scrum.
- Lista de prácticas utilizadas en la organización con técnicas y herramientas adjuntas a cada práctica.

4.3.5.3. **Fases/actividades**

Las actividades de este paso se enfocan en rediseñar el proceso tomando como base los productos de trabajo de los pasos anteriores. Dichas actividades se detallan a continuación.

A1. Eliminar prácticas que no aportan valor

Esta actividad consiste en eliminar las prácticas implementadas en la empresa que no contribuyen a la creación de valor para el cliente o la empresa, para esto se utiliza la lista de prácticas que no aportan valor obtenidas en el paso 2 a partir del análisis del valor añadido. Dichas prácticas son eliminadas de la lista de prácticas que se llevan a cabo en la organización.

A2. Cambiar prácticas específicas por prácticas ágiles

En esta actividad, las prácticas implementadas en la organización (excluyendo a las prácticas que no aportan valor) son cambiadas por prácticas ágiles de Scrum sólo en caso de que sean soportadas completamente. Para realizar este cambio se utiliza la lista de prácticas específicas del área de proceso PP que tiene una relación a prácticas equivalentes en Scrum, obtenida a partir del mapeo de prácticas en el paso 3.

A3. Sugerir técnicas y herramientas

Para esta actividad se toma como base la lista de prácticas específicas del área de PP que tiene técnicas y herramientas adjuntas (obtenida en el paso 4), a partir de esta lista se toman las técnicas y herramientas y se adjuntan a lista de prácticas implementadas en la organización, a esta última lista ya se le han eliminado prácticas que no aportan valor y se han cambiado prácticas específicas por prácticas ágiles.

A4.Rediseño del proceso

a. Identificar “focos rojos”:

En esta actividad se identifican “focos rojos”: prácticas que aportan valor de acuerdo al promedio de calificaciones de los usuarios pero dichas prácticas no aparecen en el diagrama del proceso de la organización “as is” (diagramado por el SEPG). Estas prácticas identificadas se agregan al diagrama como alertas resaltadas con color rojo.

b. Aplicación de cambios

Esta actividad se realizan los cambios y sugerencias indicados en las actividades anteriores al diagrama del proceso de la organización “as is”: eliminar las prácticas que no aportan valor, cambiar las prácticas específicas por prácticas ágiles y adjuntar técnicas y herramientas ligeras. Una vez que el diagrama este rediseñado es mostrado a los usuarios: SPEG y empleados.

4.3.5.4. Roles

En este paso los usuarios participan como visualizadores de la última actividad: rediseño del proceso (ver Tabla 23).

Tabla 23. Roles para el paso 2: Análisis del valor añadido

Actividad	Roles	
	SEPG	Empleado
Rediseño del proceso	✓	✓

4.3.5.5. Productos de trabajo

- Diagrama del proceso de la organización rediseñado.

4.4. Desarrollo de la herramienta

Para facilitar a las empresas el uso del método se desarrolla una herramienta por medio de una aplicación web, ya que entre sus principales ventajas se encuentra la facilidad de acceso desde distintas plataformas sin necesidad de descargar ni instalar software adicional, solo teniéndose como requerimiento contar con un navegador web y una conexión a internet.

Como ya se mencionó anteriormente la metodología de desarrollo elegida es UWE-UML debido a que es una completa metodología para el desarrollo de aplicaciones web basado en UML. De entre los modelos sugeridos por UWE se desarrollaron los siguientes: modelo de

requerimientos, modelo de contenido y modelo de navegación. A continuación se describe a detalle el desarrollo de cada modelo.

4.4.1. Modelo de Requerimientos

El modelado de requerimientos se realiza mediante un diagrama de casos de uso, los cuales sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios. En la Figura 12 se muestra el diagrama de casos de uso, en el cual se observa que tanto el encargado de mejoras como el usuario del proceso comparten las actividades de acceder al sistema, contestar cuestionarios y asignar valor añadido. La tarea de diagramar proceso sólo le corresponde al encargado de mejoras. Para los pasos restantes del método no hay participación de los actores encargado de mejoras y usuario del proceso.

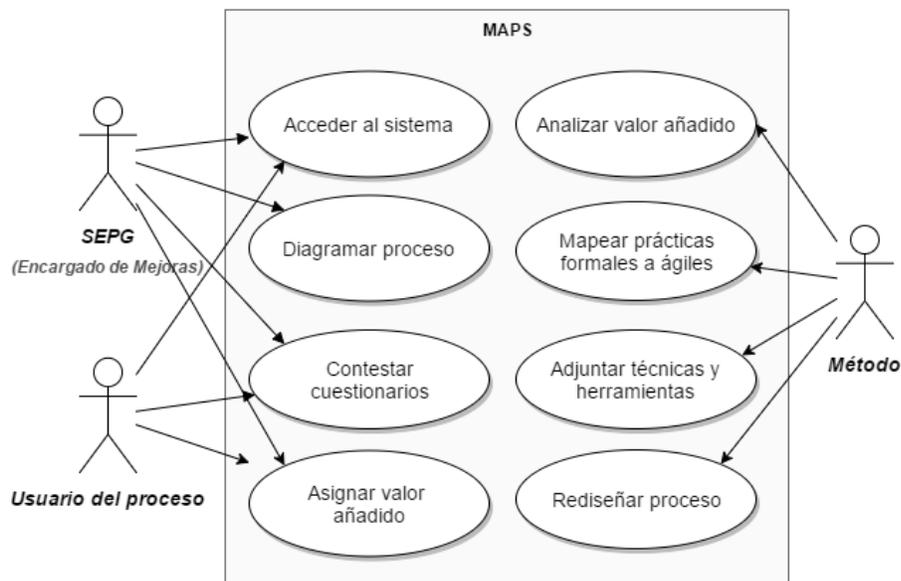


Figura 12. Diagrama de casos de uso.

A continuación en la Tabla 24 se describe más a detalle cada uno de los casos de uso mostrados en diagrama anterior.

Tabla 24. Descripción de casos de uso

ID	Nombre	Descripción
CU-1	Acceder al sistema	Los usuarios necesitan tener una cuenta personalizada con la cual puedan acceder al sistema y realizar sus actividades.
CU-2	Contestar cuestionarios	Los empleados contestan las preguntas con respecto a las prácticas implementadas en su empresa.
CU-3	Diagramar Proceso	El SEPG diagrama el proceso de la organización utilizando las prácticas implementadas en su organización.

CU-4	Asignar valor añadido	Los usuarios asignan un valor a las prácticas realizadas en la organización.
CU-5	Analizar valor añadido	El sistema analiza los valores de las prácticas, para identificar las mejores y prácticas y las que no aportan valor.
CU-6	Mapear prácticas formales a ágiles	El sistema realiza un mapeo de las prácticas realizadas en la empresa a prácticas ágiles.
CU-7	Adjuntar técnicas y herramientas	El sistema asocia técnicas y herramientas ligeras a las prácticas para ayudar en su implementación.
CU-8	Rediseñar procesos	El sistema rediseña el proceso haciendo los siguientes cambios: elimina prácticas que no aportan valor, sustituye prácticas formales por prácticas ágiles, adjunta prácticas y herramientas ligeras. El sistema muestra a los usuarios el proceso rediseñado.

4.4.2. Modelo de Contenido

El modelo de contenido es un diagrama de clases que describe la estructura estática del sistema. En la Figura 13 se muestra el diagrama de clases para el diseño de la herramienta, las clases principales se identificaron a partir del diagrama de las descripciones de los casos de uso.

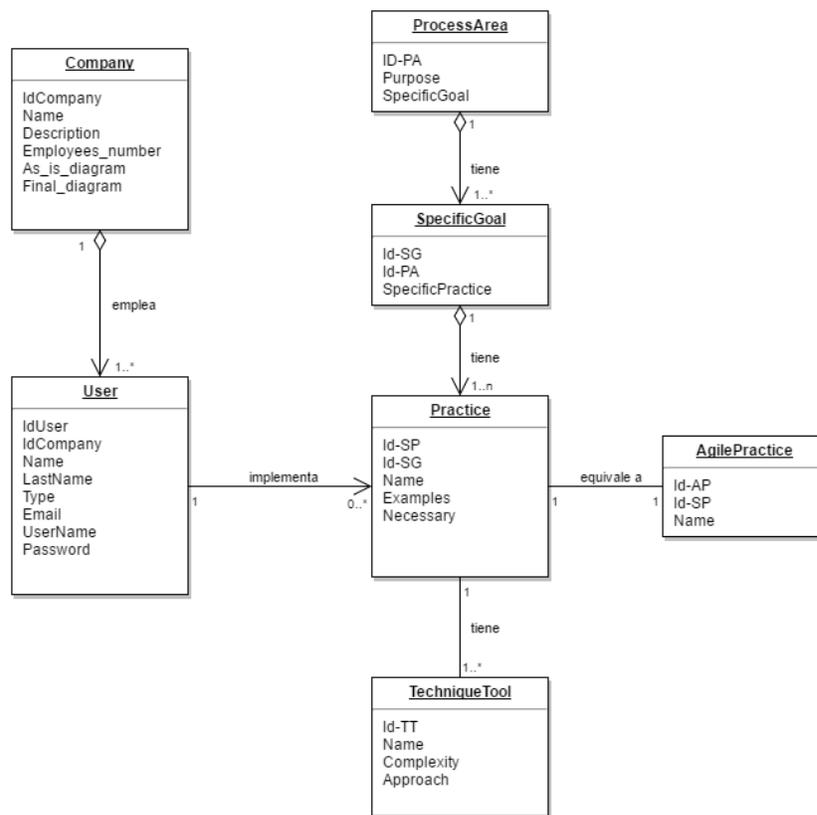


Figura 13. Diagrama de clases.

En el diagrama de clases se puede observar que la clase con más asociaciones es la clase práctica, la cual corresponde a las prácticas específicas del modelo CMMI, por lo cual hay una clase Área de Proceso que contiene Objetivos específicos mismos que a su vez contienen prácticas, las prácticas son implementadas por un usuario que pertenece a una organización, cada práctica tiene una práctica ágil equivalente y, cada práctica tiene técnicas y herramientas asociadas.

4.4.3. Modelo de Navegación

El modelo de navegación describe la manera en la que están enlazadas las páginas dentro de la aplicación web, esto por medio de nodos y enlaces. La Figura 14 muestra el modelo de navegación de la herramienta, en ella se observa que la página principal (HOME) contiene un menú de navegación en donde los usuarios pueden registrarse o iniciar sesión, el menú principal contiene enlaces hacia: la página de inicio, iniciar método (sólo cuando un usuario ha iniciado sesión), acerca del método y contacto. La página de iniciar método conduce a la ejecución del de los primeros pasos del método: modelado de la empresa y análisis del valor añadido, cuando estos pasos terminan se muestra un enlace a la página de resultados, en la cual se muestran los resultados de los pasos restantes del método.

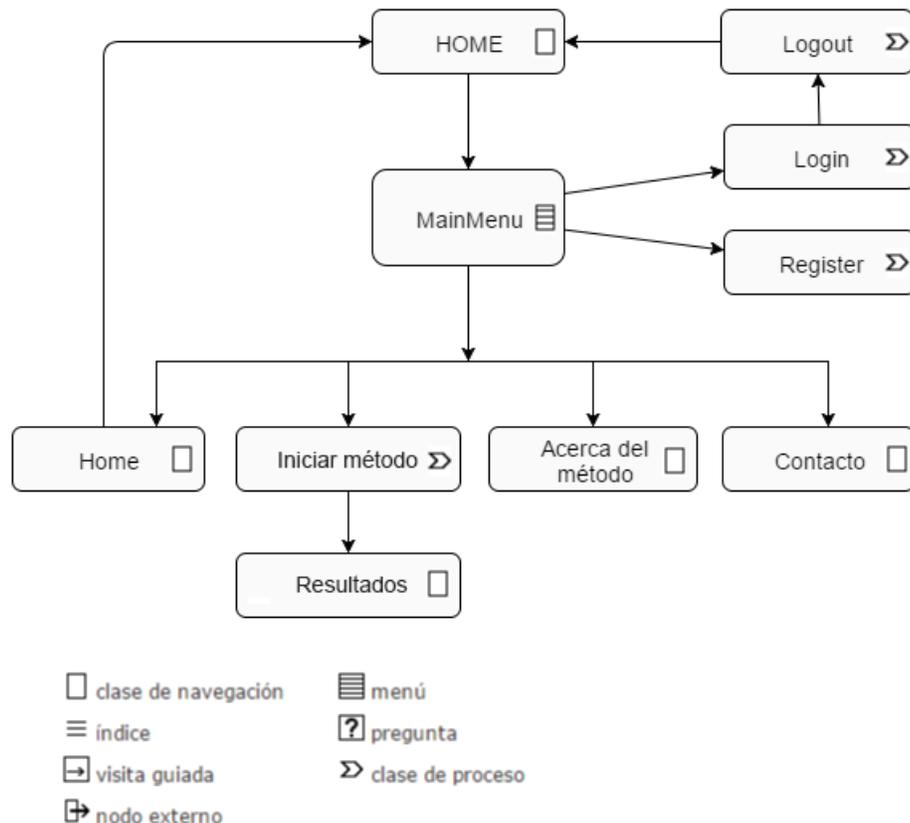


Figura 14. Modelo de navegación.

Capítulo 5. Resultados

En este capítulo se presentan los resultados de la implementación del método propuesto a través del uso de la herramienta desarrollada. Este capítulo incluye tanto la descripción de algunas de las principales pantallas de la herramienta, así como los casos de estudio diseñados y realizados para validar el método propuesto y la herramienta.

5.1. Herramienta para implementación del método propuesto

A continuación se muestran las imágenes principales de la herramienta desarrollada para facilitar el uso del método.

La Figura 15 muestra la pantalla inicial de la herramienta antes del registro de usuarios, como se observa en la figura contienen una barra de menú en la parte superior donde muestra información acerca del método y contacto, en la parte derecha del menú se encuentra el enlace iniciar sesión para cuando ya se cuenta con una cuenta, en el centro de la interfaz se encuentra el título de la aplicación y un botón para registrarse.



Figura 15. Interfaz de inicio (antes de registrarse o iniciar sesión).

Cuando un usuario selecciona la opción para registrarse aparece la interfaz de la Figura 16, donde además de solicitar datos generales como nombre, email y contraseña se pide que se seleccione la empresa a la que el usuario pertenece y su rol; que puede ser encargado de mejoras o usuario del proceso.

The screenshot shows a registration form titled "Registrarse". It contains the following elements from top to bottom: a name input field with a person icon, an email input field with "email@example.com", a company selection dropdown menu with "Seleccione su empresa:" and a blue link "Agregar empresa" below it, a role selection dropdown menu with "Seleccione su rol:", a password input field with "Contraseña (mínimo 8 caracteres)", a confirm password input field with "Confirmar contraseña", a blue "Registrarse" button, and a "Log in" link.

Figura 16. Interfaz para el registro de cuentas de usuario.

Si la empresa a la que pertenece el usuario no aparece en la lista desplegable, entonces puede agregarla seleccionando el enlace “Agregar empresa”, entonces es redirigido hacia la interfaz de la Figura 17 donde debe llenar los datos de su empresa, el número de muestra es calculado automáticamente a partir del número de emulados de la empresa.

The screenshot shows a form titled "Nueva empresa". It contains the following elements from top to bottom: a name input field with a list icon, a description input field with a list icon and a slash icon, a dropdown menu with "Experiencia en el uso de procesos", another dropdown menu with "Implementaciones de mejora (SPI)", a dropdown menu with "50", a "Muestra:" field with the value "12", a blue "Agregar empresa" button, and a "Back" link.

Figura 17. Interfaz para agregar una nueva empresa..

Una vez que un usuario accede a la aplicación por medio de su cuenta aparece una nueva interfaz de inicio (ver Figura 18). En el menú de la aplicación aparece una nueva opción: “iniciar método” el cual conduce a la ejecución de los primeros pasos del método. También en esta interfaz se muestra una breve descripción del método y un video que demuestra la manera en la que funciona la aplicación web, abajo del video aparece un botón con la opción para iniciar el método (funciona de la misma manera que su homólogo en el menú superior).

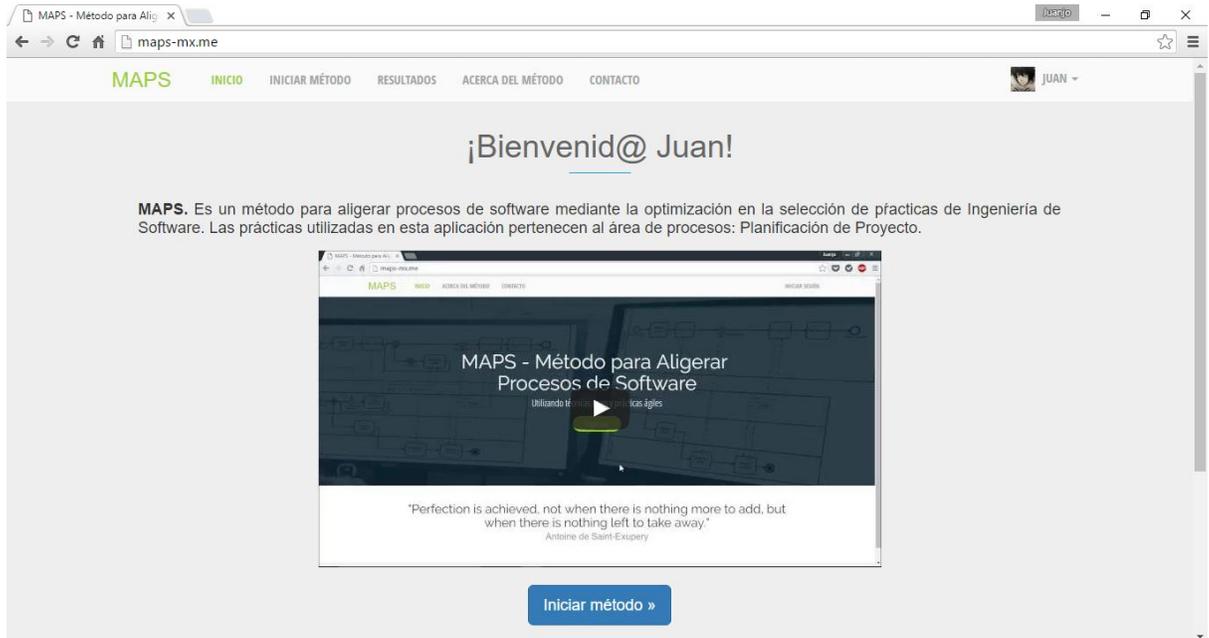


Figura 18. Interfaz de inicio (después de iniciar sesión).

En el momento que el usuario selecciona la opción “iniciar método” aparece una interfaz para la identificación de prácticas (ver

Figura 19) donde van mostrando las preguntas del cuestionario diseñado en la descripción del paso 1 del método: Modelado del proceso de la empresa, en la parte derecha aparecen las opciones disponibles: Sí, No y No sabe. También en esta interfaz se realiza la captura del valor añadido, utilizando el sistema de calificación por estrellas para asignar un valor a cada práctica.

Identificación de Prácticas

13. ¿Se concilian los niveles de trabajo y de recursos?

Ejemplo de productos de trabajo:
 * Métodos y parámetros de estimación correspondientes modificados (p. ej., mejores herramientas, uso de productos comerciales).
 * Presupuestos renegociados.
 * Calendarios modificados.
 * Lista de requisitos modificada.

Sí

No

No sabe

Califica esta práctica:

★★★★★

← Anterior
Siguiente →

Figura 19. Interfaz para la identificación de prácticas y captura del valor añadido.

Además de lo mostrado en la

Figura 19, cuando el usuario es encargado de mejoras (SEPG) tiene la opción de modelar el proceso de la empresa. El modelado de la empresa se realiza mediante la herramienta bpmn.io, la cual permite el modelado de diagramas BPMN 2.0 en la web. La Figura 20 muestra la interfaz para modelar el proceso de la empresa, las prácticas son agregadas automáticamente después de que el usuario responde afirmativamente a cada pregunta y el usuario puede realizar cambios directamente en el diagrama como mover actividades, agregar compuertas, agregar carriles, entre otras.

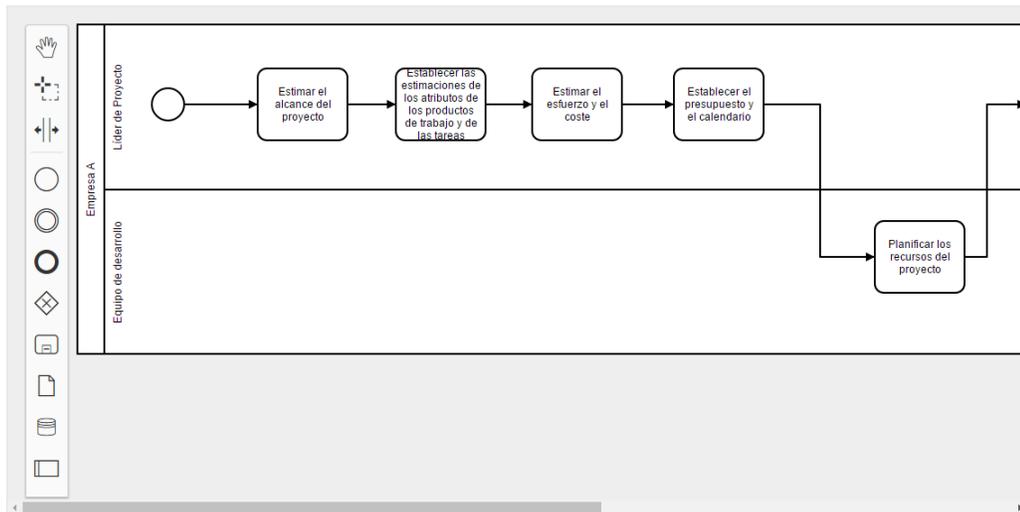


Figura 20. Herramienta para modelar el proceso de la empresa.

Cuando la cantidad de usuarios que han realizado los primeros pasos del método es mayor o igual a la muestra requerida, aparece un nuevo menú con los resultados, el cual conduce a la interfaz mostrada en la Figura 21, como se observan en la figura, los resultados están agrupados en secciones: análisis del valor añadido, mapeo a prácticas ágiles y sugerencia de técnicas y herramientas, de las cuales cada una despliega los resultados para cada paso del método. Más abajo se muestra el diagrama del proceso de la empresa con los cambios sugeridos, aparecen en color rojo arriba del diagrama las prácticas que aportan valor pero no fueron modeladas en el diagrama inicial, las prácticas que tienen una etiqueta azul de práctica ágil son las prácticas que se sugiere sean cambiadas por prácticas ágiles.



Figura 21. Interfaz con los resultados de la herramienta.

5.2. Casos de estudio

Un caso de estudio es un método empírico que investiga un fenómeno contemporáneo dentro de su contexto real, especialmente cuando los límites entre el fenómeno y el contexto no son claramente evidentes (Yin, 2002).

El caso de estudio es adecuado para muchos tipos de investigación en Ingeniería de Software, debido a que los objetos de estudio son fenómenos contemporáneos que son difíciles de estudiar de forma aislada. Los casos de estudio no generan los mismo resultados (como lo hacen los experimentos controlados), pero proporcionan una comprensión más profunda de los fenómenos que se estudian (Runeson & Höst, 2009).

La implementación de un caso de estudio consta de 5 pasos principales, los cuales se muestran a continuación (Runeson & Höst, 2009):



Figura 22. Pasos para implementar un caso de estudio.

En las siguientes secciones se muestran las actividades desarrolladas para cada paso para la implementación de los casos de estudio.

5.2.1. Diseño y planificación de los casos de estudio

La planificación de un caso de estudio es crucial para que éste tenga éxito, el diseño y planificación de un caso de estudio debe contener los siguientes elementos:

- Objetivo - ¿Qué se quiere lograr?
- El objeto de estudio - ¿Qué es lo que se estudia?
- Teoría - Marco de referencia
- Preguntas de investigación - ¿Qué se quiere conocer?
- Métodos – ¿Cómo recoger los datos?

A continuación se muestra el diseño y planificación para los casos de estudio.

5.2.1.1. **Objetivo de los casos de estudio**

Evaluar la viabilidad del método para aligerar procesos de software así como la herramienta que facilita su uso.

5.2.1.2. **Objeto de estudio**

El caso de estudio se centra el estudio de 2 empresas de desarrollo de software que utilicen la herramienta que facilita el uso del método con las siguientes características:

Tabla 25. Empresas estudiadas

Nombre	Número de empleados	Muestra de empleados	Característica
Empresa A	13	8	Sin experiencia en procesos de software.
Empresa B	50	12	Con experiencia en procesos de software.

5.2.1.3. **Marco de referencia**

Una de las preocupaciones más importantes en la industria de software es el desarrollo de productos de software con el uso óptimo de recursos, tiempo y costes, es decir, una empresa de software necesita ser eficiente. Una forma de lograr esta eficiencia es la optimización de sus procesos mediante su aligeramiento.

Asociado a lo anterior, en organizaciones con procesos definidos basados en la implementación de modelos y estándares como CMMI que han adoptado una cultura de mejora continua, un camino factible para lograr esta mejora es la optimización de sus procesos a través de su aligeramiento.

Con base en lo mencionado anteriormente, se propone un método para aligerar procesos tomando como bases tres estrategias: (1) la combinación de prácticas formales y prácticas

ágiles, (2) el uso del principio Lean: analizar prácticas que no son de valor añadido y (3) la identificación de las mejores prácticas de la organización.

5.2.1.4. Preguntas de investigación

- P1. ¿El método permite aligerar procesos de software mediante la optimización en la selección de prácticas de Ingeniería de Software?
- P2. ¿Las estrategias que utiliza el método (combinación de prácticas formales con prácticas ágiles, identificación de las mejores prácticas de la organización, uso de principios Lean) son las adecuadas para el aligeramiento de procesos de software?
- P3. ¿La forma en que se realiza la selección de prácticas para aligerar el proceso de la organización es adecuada?
- P4. ¿La herramienta es eficiente, fácil de utilizar y, por lo tanto, facilita la implementación del método?

5.2.1.5. Métodos para la recogida de datos

El método elegido para la recogida de datos es el cuestionario, debido a que los cuestionarios son formularios que los encuestados rellenan de manera individual, sin necesidad de entrevistarlos, y que además, se pueden analizar fácilmente. Para la creación de los cuestionarios se utiliza el software de encuestas Google Forms, ya que este software permite crear formularios y analizar los resultados de una forma rápida y gratuita.

5.2.2. Preparación de la recogida de datos

La recogida de datos se realiza en dos partes: en la primera, una muestra de empleados de una empresa calculado por la herramienta en base al número total de empleados de la empresa (ver capítulo 4, sección 4.2) utiliza la herramienta y observa los resultados sugeridos por la misma. En la segunda parte, dicha muestra de empleados responde un cuestionario para la evaluación de la viabilidad del método y la herramienta.

A continuación la Tabla 26 muestra las preguntas de la encuesta, de las cuales las primeras 5 se enfocan en validar cada uno de los pasos del método para aligerar procesos, para cada pregunta de la encuesta se muestra su relación con las preguntas de investigación. La recolección de respuestas para las preguntas utiliza la escala Likert con los siguientes niveles de respuesta:

- a) *Totalmente en desacuerdo*
- b) *En desacuerdo*

c) *Ni de acuerdo ni en desacuerdo*

d) *De acuerdo*

e) *Totalmente de acuerdo*

Tabla 26. Preguntas de la encuesta

Preguntas de la encuesta	Preguntas de investigación
A. ¿Considera que se ha logrado extraer las prácticas que realmente son llevadas a cabo en la organización?	P2, P3
B. ¿Considera que se ha logrado identificar las prácticas del proceso que no contribuyen a la creación de valor para el cliente o la empresa?	P2, P3
C. ¿Considera que se ha logrado identificar las mejores prácticas de la organización?	P2, P3
D. ¿Le parecen adecuadas las prácticas ágiles sugeridas para ser implementadas en tu organización?	P2, P3
E. ¿Le parecen adecuadas las técnicas y herramientas sugeridas para facilitar la implementación de prácticas en tu organización?	P2, P3
F. ¿Consideras que puede ser más eficiente el proceso rediseñado de acuerdo a las características particulares de la organización?	P1, P2, P3
Preguntas sobre la herramienta	
G. Facilidad de entendimiento: <ul style="list-style-type: none">• Se reduce el esfuerzo para realizar acciones en pocos pasos.• El contenido se transmite forma más clara y concisa.• Los mensajes son útiles para que el usuario identifique nuevas interacciones que provee la aplicación Web.• Los mensajes avisan adecuadamente al usuario sobre la acción que va a llevar a cabo.	P4
H. Eficiencia de uso: <ul style="list-style-type: none">• Los usuarios realizan sus tareas correctamente en el menor tiempo posible.• Las tareas está diseñadas para realizarse de la forma más rápida e intuitiva posible.	P4
I. Satisfacción en uso: <ul style="list-style-type: none">• La aplicación Web cubre las necesidades que te condujo a usarla.• Los resultados que obtienes tras la interacción con la aplicación Web son los deseados.• Encuentra atractivo el diseño y apariencia de la interfaz.	P4

5.2.3. Recogida de datos

Para la recogida de datos en los casos de estudio se sigue el siguiente proceso:

1. Se realiza una reunión con una muestra de empleados de la empresa en la cual se les explica brevemente el funcionamiento del método y la herramienta mediante una presentación electrónica y un video demostrativo.
2. El encargado de mejoras registra los datos de su empresa en la herramienta.

3. Los empleados se registran en la herramienta (el encargado de mejoras se registra como SEPG, mientras que el resto de empleados se registran como usuarios del proceso) y utilizan el método.
4. Una vez que todos terminan de utilizar el método se les explica los resultados mostrados por la herramienta.
5. Los empleados contestan la encuesta de validación del método y de la herramienta, diseñada en la sección anterior.
6. A partir de las encuestas contestadas se realiza el análisis de datos, el cual se describe en la siguiente sección.

5.2.4. Análisis de los datos recogidos

En esta sección se realiza un análisis de los datos recogidos en las empresas que participaron en los casos de estudio.

5.2.4.1. Caso de estudio 1

A continuación se muestra el caso de estudio de la primera empresa, que por confidencialidad será llamada Empresa A. La Tabla 27 muestra los datos generales sobre la empresa A.

Tabla 27. Datos sobre la Empresa A

Datos de la organización	
Organización	Empresa A
Descripción	Orientada a realizar proyectos internos de investigación y externos con vinculación a la industria.
Experiencia en el uso de procesos	No
Experiencia en mejora de procesos de software (SPI)	No
Tamaño	13
Participantes	7

Resultados del método

Las siguientes figuras muestran los resultados del método obtenidos para la empresa A.

En la Figura 23 y la Figura 36 se muestran los resultados del valor añadido, para cada práctica se muestra el valor y el rango en el que se clasifica de acuerdo a las calificaciones de los usuarios. Como se observa en la Figura sólo 3 prácticas se clasificaron como “mejores prácticas”, mientras que se obtuvieron 4 prácticas que no aportan valor (dentro del rango 1-25).

ID	Práctica	Valor	Rango	Análisis del valor
1	Estimar el alcance del proyecto	77.14	76-100	Mejor práctica
2	Establecer las estimaciones de los atributos de los productos de trabajo y de las tareas	28.57	26-50	Aporta poco valor
3	Definir las fases del ciclo de vida del proyecto	68.57	51-75	Aporta valor
4	Estimar el esfuerzo y el coste	47.50	26-50	Aporta poco valor
5	Establecer el presupuesto y el calendario	90.00	76-100	Mejor práctica
6	Identificar los riesgos del proyecto	25.71	26-50	Aporta poco valor
7	Planificar la gestión de los datos	25.00	1-25	No aporta valor
8	Planificar los recursos del proyecto	32.50	26-50	Aporta poco valor
9	Planificar el conocimiento y las habilidades necesarias	55.00	51-75	Aporta valor
10	Planificar la involucración de las partes interesadas	31.43	26-50	Aporta poco valor
11	Establecer el plan de proyecto	82.50	76-100	Mejor práctica
12	Revisar los planes que afectan al proyecto	17.14	1-25	No aporta valor
13	Conciliar los niveles de trabajo y de recursos	14.29	1-25	No aporta valor
14	Obtener el compromiso con el plan	22.50	1-25	No aporta valor

Figura 23. Análisis del valor añadido (Empresa A).

En la segunda sección se muestra el mapeo de prácticas formales a prácticas de Scrum (ver Figura 24), un pequeño icono de batería indica si la práctica es soportada, parcialmente soportada o no soportada. Las prácticas soportadas o parcialmente soportadas tienen una opción para ver guías sobre la implementación de la práctica. Como se observa en la figura, 8 prácticas fueron seleccionadas para realizar el mapeo y sólo 1 no es soportada.

ID	Práctica	Soporte	Práctica de Scrum	Guías
2	Establecer las estimaciones de los atributos de los productos de trabajo y de las tareas		Estimar atributos mediante Planning Poker o Magic Estimation	Ver guías >
3	Definir las fases del ciclo de vida del proyecto		Utilizar el ciclo de vida de Scrum	Ver guías >
4	Estimar el esfuerzo y el coste		Estimar el esfuerzo de las historias de usuario	Ver guías >
6	Identificar los riesgos del proyecto		Identificar lista de impedimentos	Ver guías >
8	Planificar los recursos del proyecto		Establecer el Equipo Scrum	Ver guías >
9	Planificar el conocimiento y las habilidades necesarias		No soportada	Ver guías >
10	Planificar la involucración de las partes interesadas		Definir roles y responsabilidades al principio y fin del Sprint	Ver guías >
14	Obtener el compromiso con el plan		Obtener el compromiso con el plan durante la planificación de Sprints	Ver guías >

Figura 24. Mapeo de prácticas formales a ágiles (Empresa A).

La tercera sección muestra las sugerencias de técnicas y herramientas para cada las prácticas con calificaciones altas (consideradas como mejores prácticas de la organización), en el caso de la empresa A se identificaron 3. También se realizan sugerencias para la práctica que no fue soportada por prácticas ágiles (ver Figura 25Figura 38).

ID	Práctica	Técnica o herramienta
1	Estimar el alcance del proyecto	<ul style="list-style-type: none"> Plantillas WBS de proyectos previos Información Histórica Descomposición que comprende la subdivisión en entregables y subentregables
5	Establecer el presupuesto y el calendario	<ul style="list-style-type: none"> Métodos de diagramas de precedencia (PDM) Métodos de diagramación de flechas (ADM) Métodos de diagramación condicional (Técnica de revisión y evaluación gráfica GERT) Método de ruta crítica (CPM) Técnicas de revisión y evaluación de Programas (PERT) Análisis What-If Gráficas de Gantt
9	Planificar el conocimiento y las habilidades necesarias	<ul style="list-style-type: none"> Actividades de construcción de equipos Habilidades de gestión general Sistemas de reconocimiento y recompensas Disposición Información histórica Personal y nuevas contrataciones Formación interna Formación externa Adquisición de habilidades externas Sistemas de reconocimiento y recompensas
11	Establecer el plan de proyecto	<ul style="list-style-type: none"> Metodología de planificación del proyecto Conocimiento y habilidades del la parte interesada Sistemas de Gestión del proyecto (PMIS) Gestión del valor ganado (EVM) Formas estandarizadas o plantillas Sistemas de Gestión del proyecto (PMIS) Simulaciones

Figura 25. Sugerencia de técnicas y herramientas (Empresa A).

Por último, la herramienta muestra el proceso de la organización rediseñado (ver Figura 26). En la parte superior del diagrama aparecen en color rojo las prácticas que aportan valor pero no fueron modeladas por el SEPG en el diagrama del proceso de la empresa. Las prácticas con etiqueta azul fueron cambiadas por prácticas ágiles y las prácticas que no aportan valor fueron eliminadas del diagrama, por ejemplo, la práctica 7: Planificar la gestión de los datos, fue eliminada del diagrama (originalmente se encontraba entre las prácticas: Identificar lista de impedimentos y Establecer el equipo Scrum).

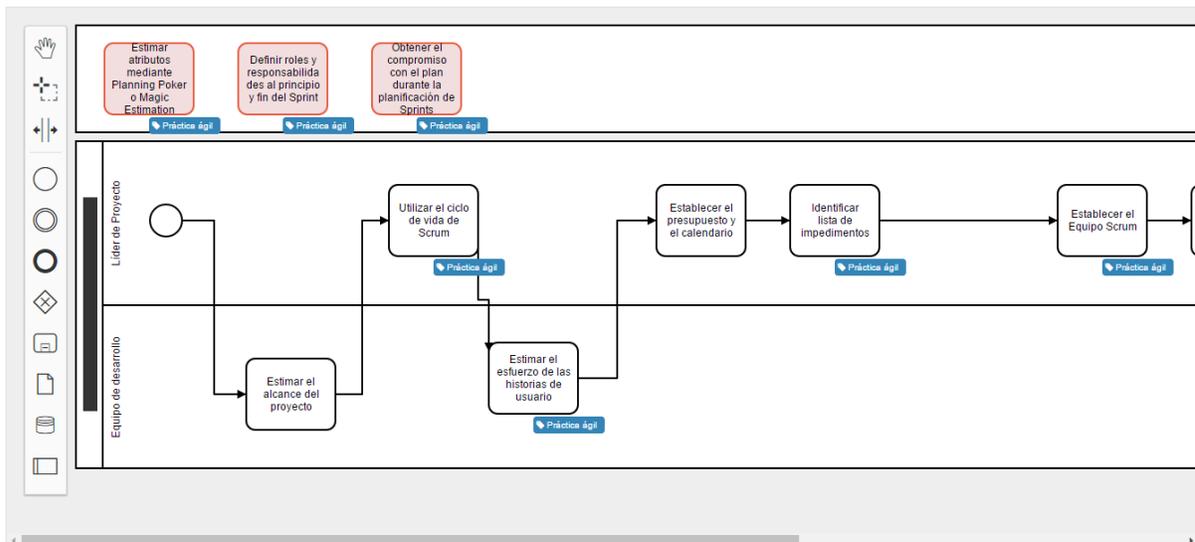


Figura 26. Proceso rediseñado (Empresa A).

Resultados de la encuesta de validación

A continuación se muestra el análisis de las respuestas de la encuesta de validación para la Empresa A.

A. ¿Considera que se ha logrado extraer las prácticas que realmente son llevadas a cabo en la organización?

Como se puede observar en el gráfico de la Figura 27, 3 encuestados (43%) están de acuerdo con la extracción de prácticas, 3 encuestados (43%) tienen una opinión neutra y 1 encuestado (14%) está en desacuerdo.

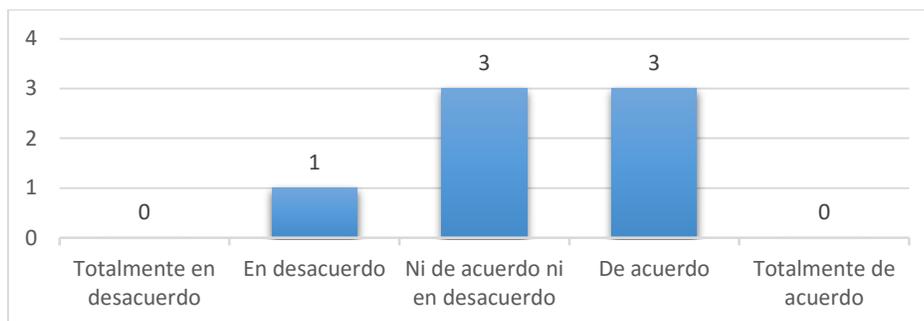


Figura 27. Respuestas de los empleados de la Empresa A para la pregunta A.

B. ¿Considera que se ha logrado identificar las prácticas del proceso que no contribuyen a la creación de valor para el cliente o la empresa?

Con respecto a la identificación de prácticas que no aportan valor, 3 encuestados (43%) está de acuerdo, 3 encuestados (43%) no está de acuerdo ni en desacuerdo y sólo 1 encuestado (14%) está en desacuerdo (ver Figura 28).

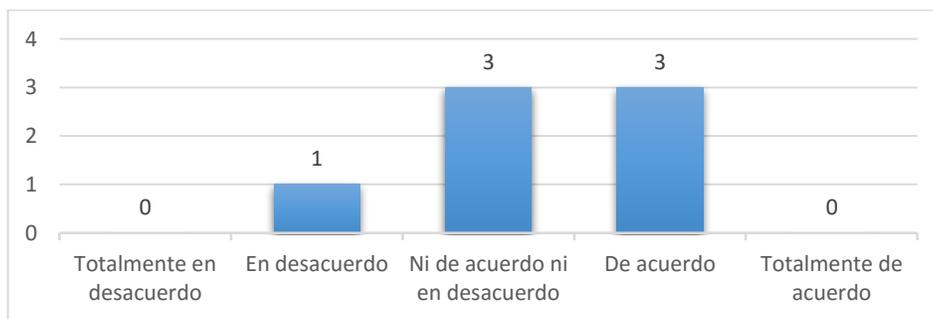


Figura 28. Respuestas de los empleados de la Empresa A para la pregunta B.

C. ¿Considera que se ha logrado identificar las mejores prácticas de la organización?

Con relación a la identificación de las mejores prácticas de la organización, 3 encuestados (43%) concuerdan con que la herramienta permite la identificación de las mejores prácticas, 3

encuestados (43%) tienen una opinión neutra y sólo 1 encuestado (14%) está en desacuerdo (Ver Figura 29).

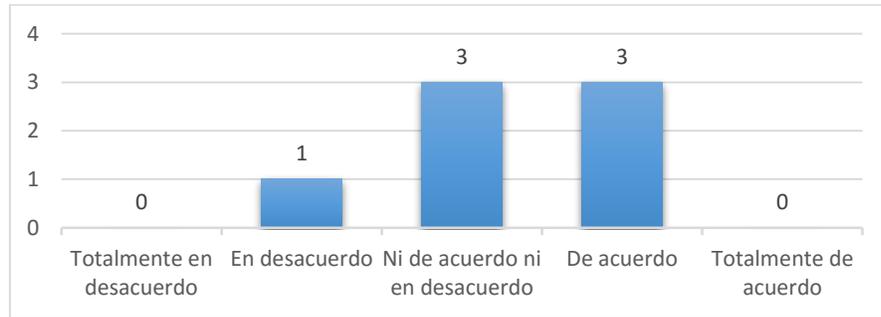


Figura 29. Respuestas de los empleados de la Empresa A para la pregunta C.

D. ¿Le parecen adecuadas las prácticas ágiles sugeridas para ser implementadas en su organización?

Esta pregunta analiza si el mapeo de prácticas formales a prácticas ágiles (paso 3 del método) es correcto, 5 de los encuestados (71%) concuerda con el mapeo, de los cuales 2 encuestados (28%) están de acuerdo y 3 encuestados (43%) totalmente de acuerdo, 2 encuestados (29%) tienen una opinión neutra y no hay encuestados en desacuerdo (Ver Figura 30).

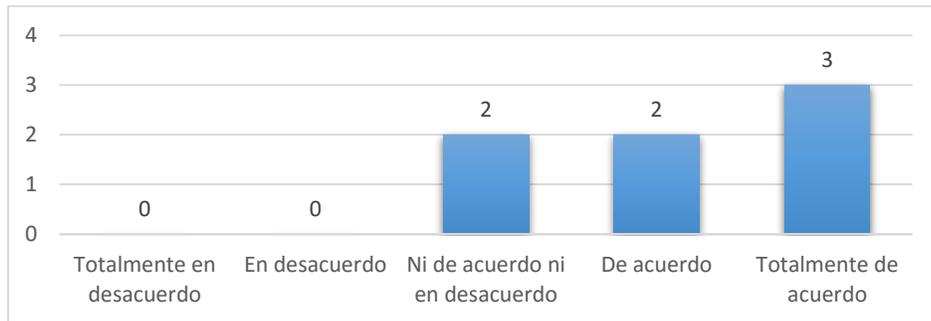


Figura 30. Respuestas de los empleados de la Empresa A para la pregunta D.

E. ¿Le parecen adecuadas las técnicas y herramientas sugeridas para facilitar la implementación de prácticas en tu organización?

Con respecto a la sugerencia de técnicas y herramientas para facilitar la implementación de las prácticas, a 3 encuestados (43%) le parecen adecuadas las técnicas y herramientas sugeridas, 3 encuestados (43%) tienen una opinión neutra y sólo 1 encuestado (14%) está en desacuerdo (Ver Figura 31).

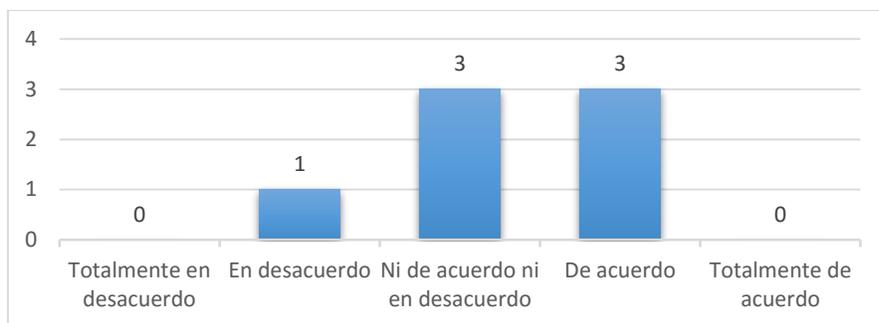


Figura 31. Respuestas de los empleados de la Empresa A para la pregunta E.

F. ¿Considera que el proceso rediseñado obtenido es más eficiente de acuerdo a las características particulares de su organización?

Esta pregunta está relacionada con las 3 primeras preguntas de investigación y además comprueba si el método es viable para la implementación de mejoras en la organización. Como se observa en la Figura 32, 4 encuestados (57%) está de acuerdo con la propuesta del proceso rediseñado, 2 encuestados (29%) tienen una opinión neutra y 1 encuestado (14%) está en desacuerdo.

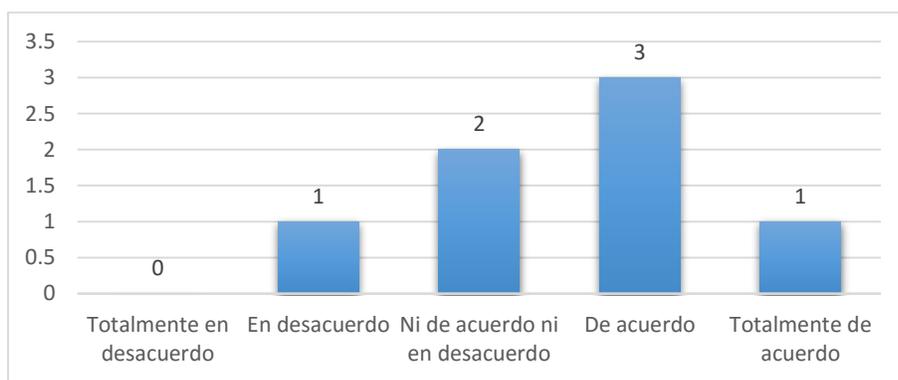


Figura 32. Respuestas de los empleados de la Empresa A para la pregunta F.

G. Facilidad de entendimiento

Para revisar facilidad de entendimiento de la herramienta se analizan los siguientes atributos (ver Figura 33):

- **Acciones mínimas:** Se reduce el esfuerzo para realizar acciones en pocos pasos. 5 de los encuestados (71%), mientras que 2 encuestados (29%) no están de acuerdo.
- **Auto-descripción:** El contenido se transmite de forma clara y concisa. 4 encuestados (57%) concuerdan con este atributo, mientras que 3 encuestados (43%) no están de acuerdo.
- **Calidad de los mensajes de actualización:** Los mensajes son útiles para que usuario identifica nuevas interacciones que provee la aplicación web. De los encuestados, 5

(71%) están de acuerdo con este atributo, mientras que 2 encuestados (29%) no está de acuerdo.

- **Calidad de los mensajes de aviso:** Los mensajes avisan adecuadamente al usuario sobre la acción que va a llevar a cabo. 4 de los encuestados (57%) concuerdan con este atributo, mientras que 3 encuestados (43%) no están de acuerdo.

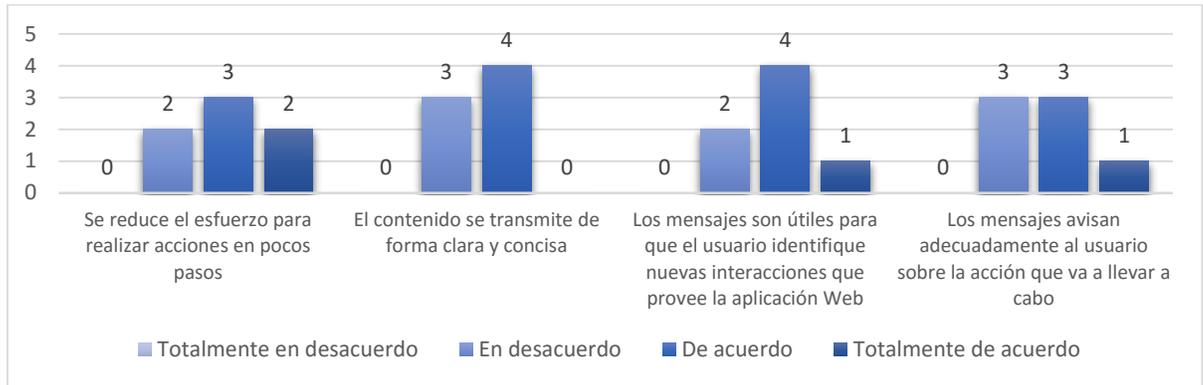


Figura 33. Respuestas de los empleados de la Empresa A para la pregunta G.

H. Eficiencia de uso:

Para revisar eficiencia de uso de la herramienta se analizan los siguientes atributos (ver Figura 34):

- **Tiempo para completar las tareas:** Los usuarios realizan sus tareas correctamente en el menor tiempo posible. De los encuestados, 5 encuestados (71%) están de acuerdo con este atributo, sólo 2 encuestados (29%) no están de acuerdo.
- **Carga de las tareas:** Las tareas están diseñadas para realizarse de la forma más rápida e intuitiva posible. 4 de los encuestados (57%), están de acuerdo, mientras que 3 encuestados (42%) no están de acuerdo.

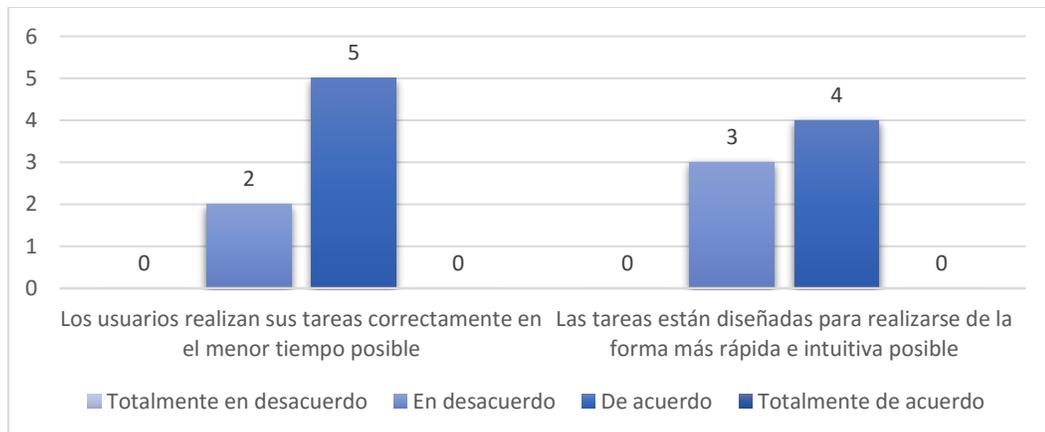


Figura 34. Respuestas de los empleados de la Empresa A para la pregunta H.

I. Satisfacción de uso:

Para revisar la satisfacción de uso de la herramienta se analizan los siguientes atributos (ver Figura 35):

- **Utilidad percibida:** La aplicación Web cubre las necesidades que te condujo a usarla. De los encuestados, 5 (71%) están de acuerdo con este atributo, mientras que 2 encuestados (29%) están en desacuerdo.
- **Calidad de los resultados:** Los resultados que obtiene tras la interacción con la aplicación Web son los deseados. De los encuestados, 6 (85%) están de acuerdo, sólo 1 encuestado se encuentra en desacuerdo.
- **Atracción subjetiva percibida:** Encuentra atractivo el diseño y la apariencia de la interfaz. 6 de los encuestados (85%) concuerdan con este punto, sólo 1 encuestado (8%) no encuentra atractivo el diseño y la apariencia de la interfaz.

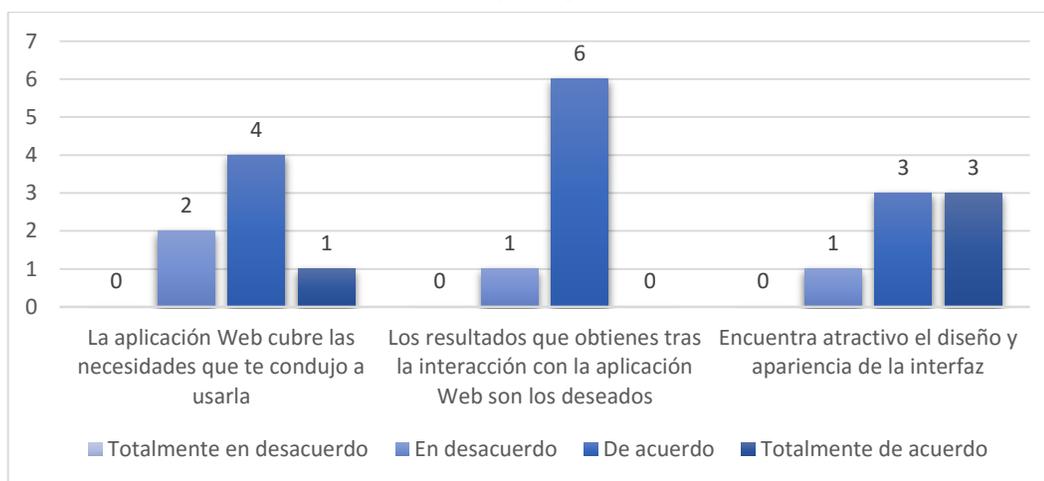


Figura 35. Respuestas de los empleados de la Empresa A para la pregunta I.

5.2.4.2. Caso de estudio 2

A continuación se muestra el caso de estudio de la segunda organización, que por confidencialidad será llamada Empresa B. La Tabla 28 muestra los datos generales sobre la empresa B.

Tabla 28. Datos sobre la Empresa B

Datos de la organización	
Organización	Empresa B
Descripción	Dedicada al desarrollo de sistemas y soluciones informáticas, ofrece además servicios paralelos de proveeduría de sistemas (SaaS), implantación, capacitación y asesoría.
Experiencia en el uso de procesos	Sí

Experiencia en mejora de procesos de software (SPI)	Sí
Tamaño	50
Participantes	12

Resultados del método

Las siguientes figuras muestran los resultados del método obtenidos para la empresa B.

En la Figura 36 se muestran los resultados del valor añadido, para cada práctica se muestra el valor y el rango en el que se clasifica de acuerdo a las calificaciones de los usuarios. Como se observa en la figura la mayoría de las prácticas obtuvieron calificaciones mayores al 50% y no hubo prácticas que no aportaran valor (dentro del rango 1-25).

ID	Práctica	Valor	Rango	Análisis del valor
1	Estimar el alcance del proyecto	51.67	51-75	Aporta valor
2	Establecer las estimaciones de los atributos de los productos de trabajo y de las tareas	58.33	51-75	Aporta valor
3	Definir las fases del ciclo de vida del proyecto	83.64	76-100	Mejor práctica
4	Estimar el esfuerzo y el coste	58.33	51-75	Aporta valor
5	Establecer el presupuesto y el calendario	58.33	51-75	Aporta valor
6	Identificar los riesgos del proyecto	51.67	51-75	Aporta valor
7	Planificar la gestión de los datos	46.67	26-50	Aporta poco valor
8	Planificar los recursos del proyecto	41.67	26-50	Aporta poco valor
9	Planificar el conocimiento y las habilidades necesarias	86.67	76-100	Mejor práctica
10	Planificar la involucración de las partes interesadas	32.73	26-50	Aporta poco valor
11	Establecer el plan de proyecto	84.00	76-100	Mejor práctica
12	Revisar los planes que afectan al proyecto	50.00	26-50	Aporta poco valor
13	Conciliar los niveles de trabajo y de recursos	46.67	26-50	Aporta poco valor
14	Obtener el compromiso con el plan	58.33	51-75	Aporta valor

Figura 36. Análisis del valor añadido (Empresa B).

En la segunda sección se muestra el mapeo de prácticas formales a prácticas de Scrum (ver Figura 37), un pequeño icono de batería indica si la práctica es soportada, parcialmente soportada o no soportada. Las prácticas soportadas o parcialmente soportadas tienen una opción para ver guías sobre la implementación de la práctica. Como se observa en la figura, 11 prácticas fueron seleccionadas para realizar el mapeo y sólo 2 no son soportadas.

ID	Práctica	Soporte	Práctica de Scrum	Guías
1	Estimar el alcance del proyecto	<input type="checkbox"/>	No soportada	Ver guías
2	Establecer las estimaciones de los atributos de los productos de trabajo y de las tareas	<input checked="" type="checkbox"/>	Estimar atributos mediante Planning Poker o Magic Estimation	Ver guías >
4	Estimar el esfuerzo y el coste	<input checked="" type="checkbox"/>	Estimar el esfuerzo de las historias de usuario	Ver guías >
5	Establecer el presupuesto y el calendario	<input checked="" type="checkbox"/>	Establecer Sprints	Ver guías >
6	Identificar los riesgos del proyecto	<input checked="" type="checkbox"/>	Identificar lista de impedimentos	Ver guías >
7	Planificar la gestión de los datos	<input type="checkbox"/>	No soportada	Ver guías
8	Planificar los recursos del proyecto	<input checked="" type="checkbox"/>	Establecer el Equipo Scrum	Ver guías >
10	Planificar la involucración de las partes interesadas	<input checked="" type="checkbox"/>	Definir roles y responsabilidades al principio y fin del Sprint	Ver guías >
12	Revisar los planes que afectan al proyecto	<input checked="" type="checkbox"/>	Revisar los planes que afectan al proyecto en la revisión de Sprint	Ver guías >
13	Conciliar los niveles de trabajo y de recursos	<input checked="" type="checkbox"/>	Conciliar los niveles de trabajo y de recursos en la planificación de Sprints	Ver guías >
14	Obtener el compromiso con el plan	<input checked="" type="checkbox"/>	Obtener el compromiso con el plan durante la planificación de Sprints	Ver guías >

Figura 37. Mapeo de prácticas formales a ágiles (Empresa B).

La tercera sección muestra las sugerencias de técnicas y herramientas para cada las prácticas con calificaciones altas (consideradas como mejores prácticas de la organización), también se realizan sugerencias de técnicas y herramientas para las prácticas no soportadas por prácticas ágiles (ver Figura 38).

ID	Práctica	Técnica o herramienta
1	Estimar el alcance del proyecto	<ul style="list-style-type: none"> Plantillas WBS de proyectos previos Información Histórica Descomposición que comprende la subdivisión en entregables y subentregables
3	Definir las fases del ciclo de vida del proyecto	<ul style="list-style-type: none"> Subdividir el proyecto en componentes más pequeños y manejables para soporte de las actividades del proyecto (planificación, ejecución, control y término) Plantilla WBS pueden ser usadas como plantillas para nuevos proyectos
7	Planificar la gestión de los datos	<ul style="list-style-type: none"> Habilidades de Comunicación Sistemas manuales de archivos Bases de datos electrónicas Software de gestión de proyectos Formas estandarizadas o plantillas
9	Planificar el conocimiento y las habilidades necesarias	<ul style="list-style-type: none"> Actividades de construcción de equipos Habilidades de gestión general Sistemas de reconocimiento y recompensas Disposición Información histórica Personal y nuevas contrataciones Formación interna Formación externa Adquisición de habilidades externas Sistemas de reconocimiento y recompensas
11	Establecer el plan de proyecto	<ul style="list-style-type: none"> Metodología de planificación del proyecto Conocimiento y habilidades del la parte interesada Sistemas de Gestión del proyecto (PMIS) Gestión del valor ganado (EVM) Formas estandarizadas o plantillas Sistemas de Gestión del proyecto (PMIS) Simulaciones

Figura 38. Mapeo de prácticas formales a ágiles (Empresa B).

Por último, la herramienta muestra el proceso de la organización rediseñado (ver Figura 39). A diferencia de la empresa A, para la empresa B no hay prácticas que aporten valor faltantes en el diagrama del proceso (ya que todas las prácticas fueron diagramadas por el SEPG), así

mismo tampoco hay prácticas eliminadas en el proceso rediseñado. Sin embargo, la mayoría de las prácticas fueron cambiadas por prácticas ágiles.

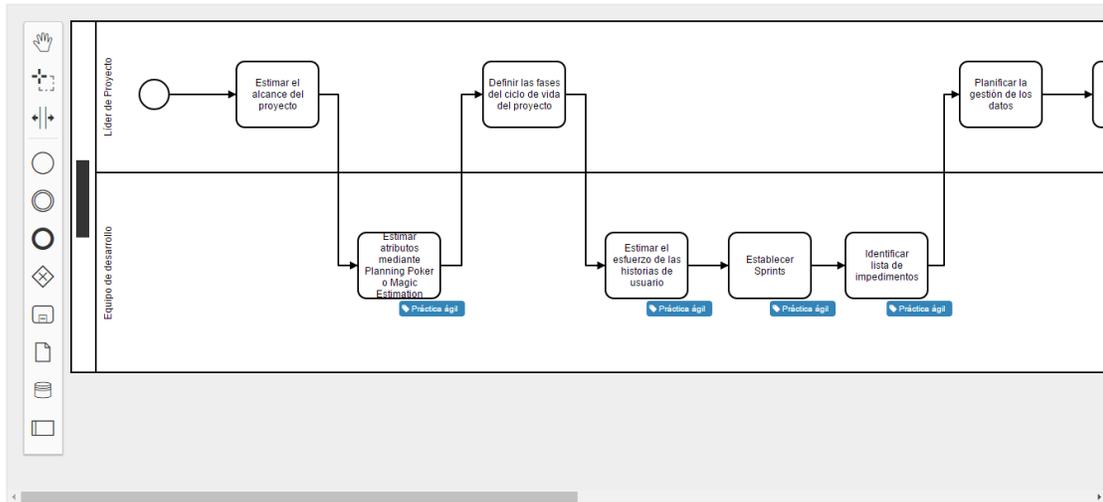


Figura 39. Proceso rediseñado (Empresa B).

Resultados de la encuesta de validación

A continuación se muestra el análisis de las respuestas de la encuesta de validación para la Empresa B.

A. ¿Considera que se ha logrado extraer las prácticas que realmente son llevadas a cabo en la organización?

Como se puede observar en el gráfico de la Figura 40, de los encuestados, 10 (83%) están de acuerdo con la extracción de prácticas, 2 encuestados (17%) tienen una opinión neutra y no hay encuestados que estén en desacuerdo.

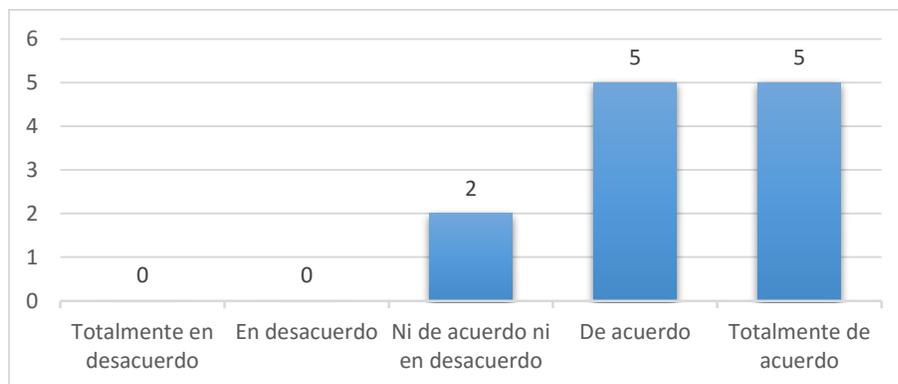


Figura 40. Respuestas de los empleados de la Empresa B para la pregunta A.

B. ¿Considera que se ha logrado identificar las prácticas del proceso que no contribuyen a la creación de valor para el cliente o la empresa?

Con respecto a la identificación de prácticas que no aportan valor, 10 encuestados (83%) está de acuerdo, 2 encuestados (17%) no está de acuerdo ni en desacuerdo y no hay encuestados en desacuerdo (ver Figura 41).

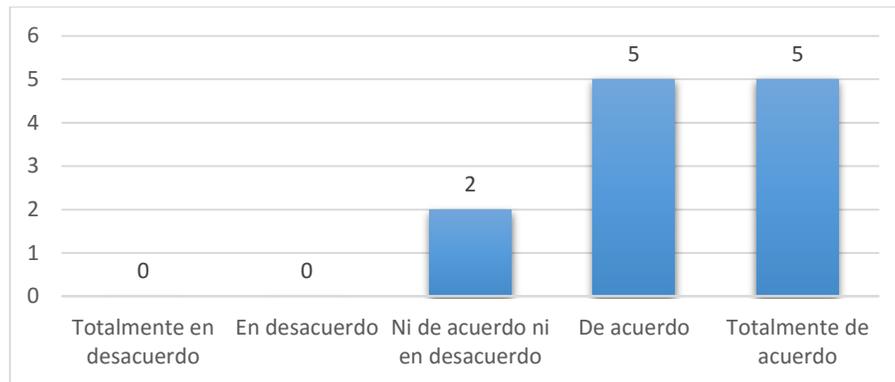


Figura 41. Respuestas de los empleados de la Empresa B para la pregunta B.

C. ¿Considera que se ha logrado identificar las mejores prácticas de la organización?

Con relación a la identificación de las mejores prácticas de la organización, 10 encuestados (83%) concuerdan con que la herramienta permite la identificación de las mejores prácticas, de dichos encuestados 6 encuestados (50%) están de acuerdo y 4 (33%) están totalmente de acuerdo, 2 encuestados tienen una opinión neutra y no hay encuestados en desacuerdo (Ver Figura 42).

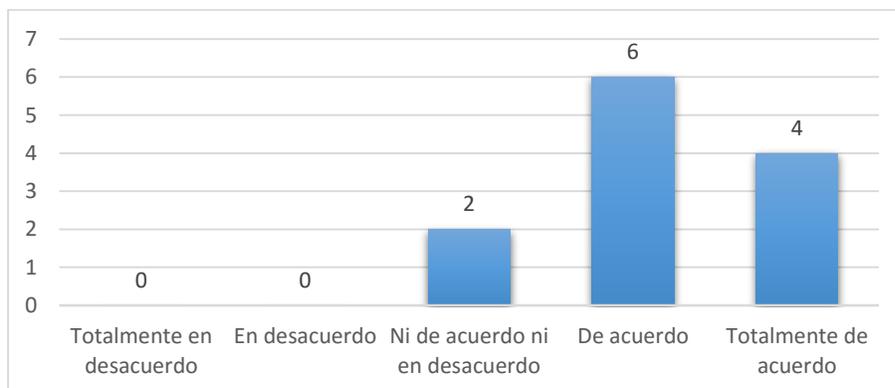


Figura 42. Respuestas de los empleados de la Empresa B para la pregunta C.

D. ¿Le parecen adecuadas las prácticas ágiles sugeridas para ser implementadas en su organización?

Esta pregunta analiza si el mapeo de prácticas formales a prácticas ágiles (paso 3 del método) es correcto, 10 de los encuestados (83%) concuerda con el mapeo, de los cuales 4 encuestados

(33%) están de acuerdo y 6 encuestados (50%) totalmente de acuerdo, 2 encuestados (17%) tienen una opinión neutra y no hay encuestados en desacuerdo (Ver Figura 43).

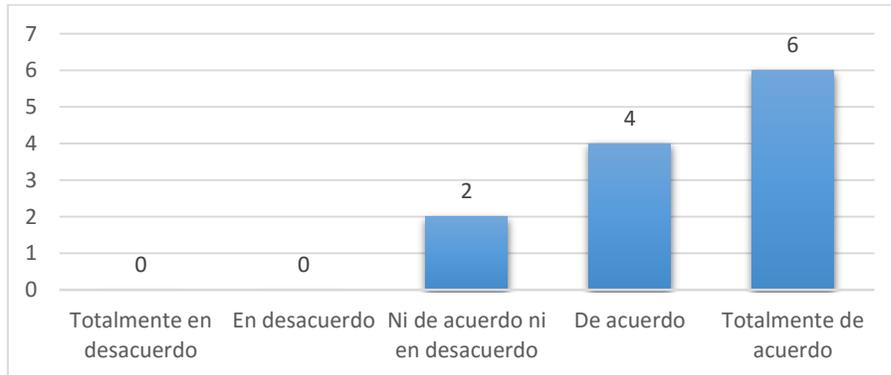


Figura 43. Respuestas de los empleados de la Empresa B para la pregunta D.

E. ¿Le parecen adecuadas las técnicas y herramientas sugeridas para facilitar la implementación de prácticas en tu organización?

Con respecto a la sugerencia de técnicas y herramientas para facilitar la implementación de las prácticas, a 11 encuestados (92%) le parecen adecuadas, de los cuales 4 (42%) está de acuerdo y 6 (50%) totalmente de acuerdo, sólo 1 encuestado tiene una opinión neutra y no hay encuestados en desacuerdo (Ver Figura 44).

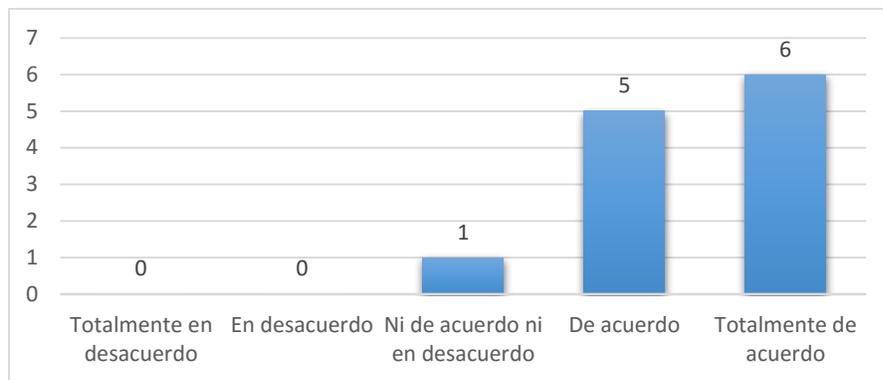


Figura 44. Respuestas de los empleados de la Empresa B para la pregunta E.

F. ¿Considera que el proceso rediseñado obtenido es más eficiente de acuerdo a las características particulares de su organización?

Esta pregunta está relacionada con las 3 primeras preguntas de investigación y además comprueba si el método es viable para la implementación de mejoras en la organización. Como se observa en la Figura 45, 9 encuestados (75%) está de acuerdo con la propuesta del proceso rediseñado, 2 encuestados (17 %) tienen una opinión neutra y 1 encuestado (8%) está en desacuerdo.

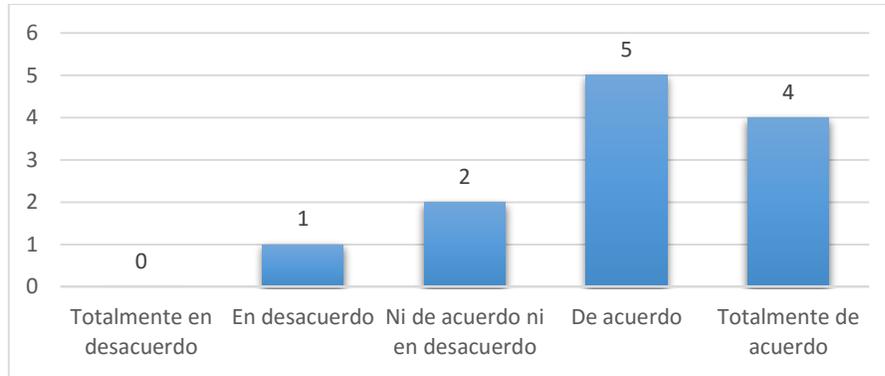


Figura 45. Respuestas de los empleados de la Empresa B para la pregunta F.

G. Facilidad de entendimiento

Para revisar facilidad de entendimiento de la herramienta se analizan los siguientes atributos (ver Figura 46):

- **Acciones mínimas:** Se reduce el esfuerzo para realizar acciones en pocos pasos. Todos los encuestados concuerdan con este atributo, 6 encuestados (50%) está de acuerdo y los otros 6 (50%) totalmente de acuerdo.
- **Auto-descripción:** El contenido se transmite de forma clara y concisa. Los 12 encuestados (100%) concuerdan con este atributo, 6 encuestados (50%) están de acuerdo y los 6 restantes (50%) están totalmente de acuerdo.
- **Calidad de los mensajes de actualización:** Los mensajes son útiles para que usuario identifica nuevas interacciones que provee la aplicación web. De los encuestados, 11 (92%) concuerda con este atributo, sólo 1 encuestado (8%) no está de acuerdo.
- **Calidad de los mensajes de aviso:** Los mensajes avisan adecuadamente al usuario sobre la acción que va a llevar a cabo. 12 de los encuestados (83%) concuerdan con este atributo, de los cuales 4 (33%) están de acuerdo y 6 (50%) totalmente de acuerdo, por otra parte, 2 encuestados (17%) no están de acuerdo.

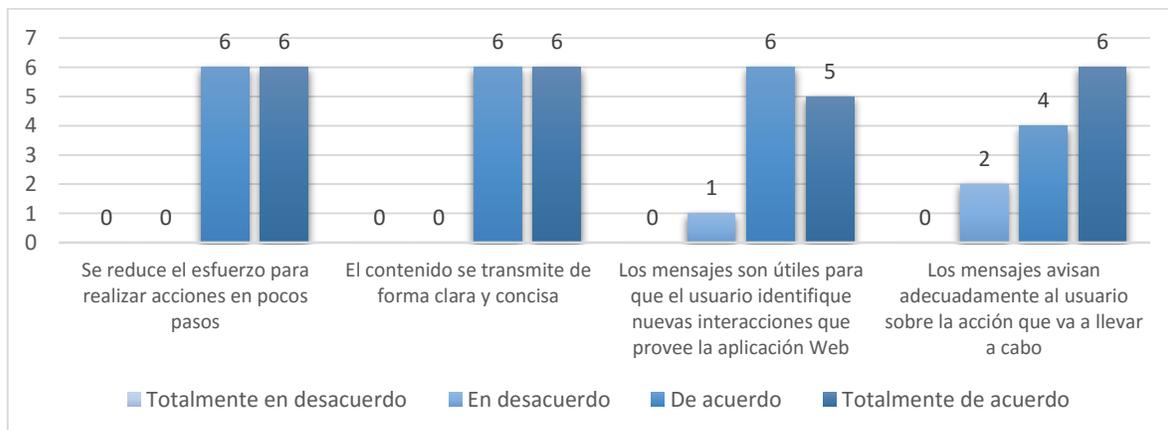


Figura 46. Respuestas de los empleados de la Empresa B para la pregunta G.

H. Eficiencia de uso:

Para revisar eficiencia de uso de la herramienta se analizan los siguientes atributos (ver Figura 47):

- **Tiempo para completar las tareas:** Los usuarios realizan sus tareas correctamente en el menor tiempo posible. Todos los encuestados concuerdan con este atributo, de los cuales 9 (75%) están de acuerdo y 3 (25%) totalmente de acuerdo.
- **Carga de las tareas:** Las tareas están diseñadas para realizarse de la forma más rápida e intuitiva posible. Nuevamente todos los encuestados concuerdan con este atributo, de los cuales 10 encuestados (83%) están desacuerdo y 2 (17%) totalmente de acuerdo.

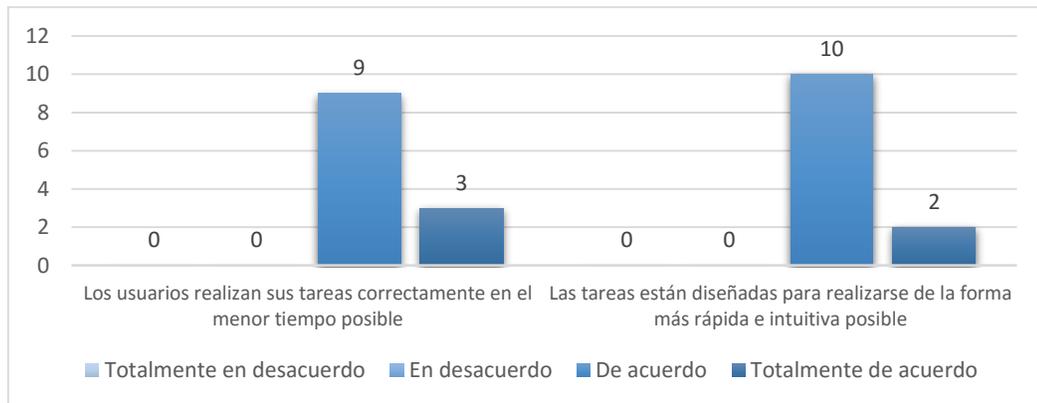


Figura 47. Respuestas de los empleados de la Empresa B para la pregunta H.

I. Satisfacción de uso:

Para revisar la satisfacción de uso de la herramienta se analizan los siguientes atributos (ver Figura 46):

- **Utilidad percibida:** La aplicación Web cubre las necesidades que te condujo a usarla. Todos los encuestados (100%) concuerdan con este atributo, de los cuales 4 (33%) está de acuerdo y 8 (67%) totalmente de acuerdo.
- **Calidad de los resultados:** Los resultados que obtiene tras la interacción con la aplicación Web son los deseados. Nuevamente todos los encuestados (100%) concuerdan con este atributo, de los cuales la mitad (50%) está de acuerdo y la otra mitad (50%) totalmente de acuerdo.
- **Atracción subjetiva percibida:** Encuentra atractivo el diseño y la apariencia de la interfaz. De los encuestados, 11 (92%) concuerdan con este punto, 3 (25%) está de acuerdo y 8 (67%) está totalmente de acuerdo, sólo 1 (8%) persona no encuentra atractivo el diseño y la apariencia de la interfaz.

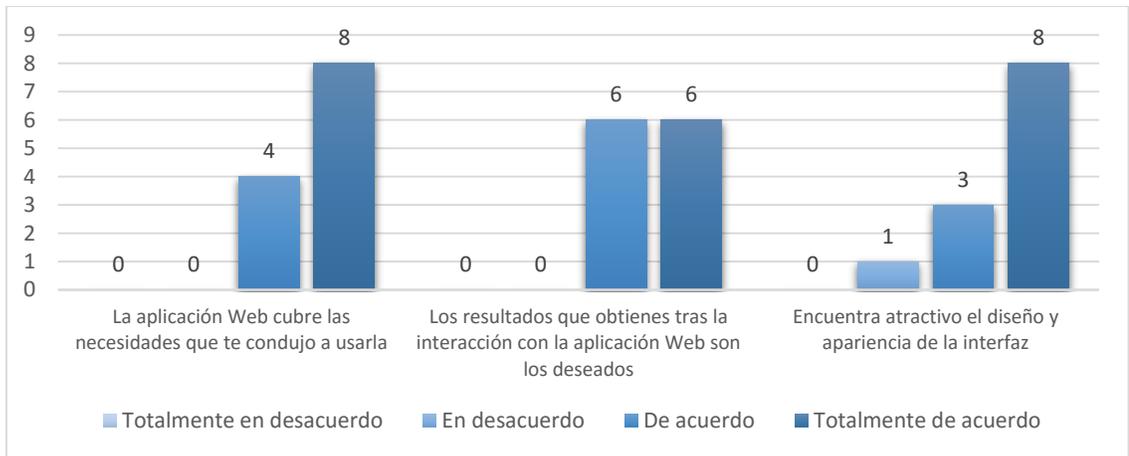


Figura 48. Respuestas de los empleados de la Empresa B para la pregunta I.

Comentarios y recomendaciones de los encuestados

A continuación se muestran los comentarios y recomendaciones de los encuestados de la empresa A:

- *“Muy buena la herramienta, para mayor aplicabilidad habría que pensar en ampliar el número de áreas en donde se puede aplicar y poder cambiar las prácticas por las verdaderamente empleadas en la empresa”.*
- *“En las prácticas que no son soportadas por metodologías ágiles podría el método dar alguna alternativa”.*
- *“Pueden proponerse o evaluarse diferentes técnicas y herramientas según las necesidades de la empresa? Incluso por áreas ya que no todas tienen el mismo proceso para realizar las mismas tareas.”*
- *“Creo que la herramienta está muy bien y puede ayudar mucho a las empresas en cuanto a mejora continua de sus mejores prácticas.”*

5.2.5. Reporte de resultados

Al observar los resultados obtenidos por las empresas participantes al utilizar la herramienta se pudo observar que la Empresa A, la cual no tienen experiencia en procesos ni en implementaciones de mejora, requirió una gran cantidad de cambios en el rediseño de su proceso: se detectaron prácticas que no aportaron valor, de las cuales la prescindibles se eliminaron, y hubo prácticas faltantes en el proceso original que de acuerdo a los demás integrantes de la empresa si se realizan y aportan valor, lo cual denota que no se tiene un consenso sobre el proceso de la empresa.

Por otra parte, la Empresa B, la cual tiene experiencia en procesos y en implementaciones de mejora, requirió menos cambios en el rediseño de su proceso: la mayoría de las prácticas obtuvieron calificaciones altas, no se encontraron prácticas que no aportaran valor ni prácticas faltantes en el proceso original, lo cual denota que hay un consenso sobre el proceso de la empresa.

A partir del análisis de los datos recogidos de las encuestas de los casos de estudio se han comprobado y validado los siguientes resultados del método y la herramienta:

- El uso del método es viable para la implementación de mejoras en la organización a través del aligeramiento de los procesos mediante la selección de prácticas, esto debido a que en ambos casos de estudios fueron más los encuestados que estuvieron de acuerdo en que el proceso rediseñado puede ser más eficiente que el proceso original de la empresa.
- Las estrategias que utiliza el método (combinación de prácticas formales con prácticas ágiles, identificación de las mejores prácticas de la organización, uso de principios Lean) son adecuadas para aligeramiento de procesos de software debido a que la mayoría de los encuestados estuvieron de acuerdo con los resultados de cada paso del método.
- De acuerdo a los encuestados, la herramienta es fácil de utilizar, eficiente y cuenta con un diseño atractivo, por lo tanto, facilita la implementación del método para aligerar procesos.

Así mismo se acepta la hipótesis planteada, ya que de acuerdo a los resultados de la pregunta F (en ambos casos de estudio), los usuarios perciben que los resultados obtenidos por el método pueden ayudar a incrementar la eficiencia del proceso de la organización

Capítulo 6. Conclusiones

En este capítulo se presentan las conclusiones obtenidas. Además, se presenta el trabajo futuro derivado de este tema de investigación, y por último, se abordan los productos académicos de esta investigación.

6.1. Conclusiones

El desarrollo de este trabajo ha resultado en la propuesta de un método para aligerar procesos de software al optimizar la selección de prácticas implementadas en una organización. Con el método propuesto se apoya a dos tipos de organizaciones: (1) empresas de software con cultura de procesos y con experiencia en implementaciones de mejoras; mostrándoles un camino factible para lograr una mejora continua de procesos y, (2) empresas de software sin cultura de procesos pero con inquietud en implementar mejoras; mostrándoles un camino para formalizar su proceso a partir de un consenso entre una muestra significativa de empleados de la empresa.

El desarrollo del método toma como base la realización de una revisión sistemática de la literatura, cuyo objetivo fue conocer el estado actual referente al aligeramiento de procesos de software, enfocándose en tres elementos clave: marcos de trabajo, métodos y metodologías, procesos de software enfocados y estrategias utilizadas. Algunos hallazgos y conclusiones de la revisión se muestran a continuación:

- Sólo se encontraron estudios sobre el aligeramiento de procesos para el modelo CMMI, no se encontró evidencia de propuestas que utilicen otros modelos o estándares importantes como el ISO/IEC 15504.
- La categoría de CMMI en la que más se centran los estudios sobre el aligeramiento de procesos es la de ingeniería, principalmente en las áreas de proceso de solución técnica, validación de software y desarrollo de requisitos, por lo que hay una falta estudios enfocados en áreas de proceso de otras categorías como como gestión de proyectos.
- En cuanto a las estrategias para aligerar procesos de software, La estrategia más utilizada para aligerar procesos de software es el uso de herramientas, sin embargo,

estas herramientas se centran en la reducción del tiempo y esfuerzo que toma la ejecución de un proceso, sin realizar un análisis que permita la optimización del mismo. Además analizando otras estrategias como: la combinación de prácticas de método formales con prácticas de desarrollo ágil, la identificación de las mejores prácticas de la organización o el uso de principios Lean, se concluye que las estrategias existentes utilizadas para aligerar procesos de software pueden ser integradas para la creación de un método que analice los procesos para su correcto aligeramiento, tal que puedan obtenerse procesos optimizados que apoyen a la mejora continua de las organizaciones.

A partir de la comparativa de trabajos relacionados se concluye que no hay ningún trabajo relacionado cuya propuesta cubra completamente más de una estrategia para aligerar procesos, tal como lo hace el método propuesto en el presente trabajo.

Con respecto a los objetivos específicos:

- Se obtuvo el estado del arte respecto al aligeramiento de procesos de software, obteniendo estrategias para aligerar procesos que sirvieron como base para el desarrollo del método.
- Se identificaron y analizaron los modelos y estándares más utilizados en la industria de software: CMMI, ISO/IEC 15504, MoProsfot e ISO/IEC 29110.
- Se implementó un catálogo de técnicas y herramientas para el paso 4 del método: sugerencia de técnicas y herramientas.
- Se desarrolló un método para aligerar procesos con base en las estrategias encontradas al obtener el estado del arte.
- De diseño y desarrolló una herramienta mediante una aplicación web para facilitar el uso del método.
- Se validó la viabilidad del método y su herramienta mediante dos casos de estudio en dos empresas (8 participantes en la empresa A y 12 en la empresa B), en ambos casos, los participantes estuvieron de acuerdo con los resultados obtenidos por el método.

6.2. Trabajo futuro

Como trabajo futuro se plantea mejorar el método con los siguientes cambios:

- Extender el método a más áreas de proceso, para no tener sólo prácticas del área de planificación del proyecto.
- Agregar modelos o estándares para que los usuarios de una empresa puedan modelar su proceso e identificar prácticas a partir del modelo que utilizan en su organización.

- Realizar un análisis de dependencias entre las prácticas para mejorar la selección de prácticas.
- Integrar el método a una plataforma para la mejora de procesos de software.

6.3. Logros académicos

A continuación se muestran los logros académicos obtenidos a partir de la presente investigación.

6.3.1. Productos académicos

Miramontes, J., Muñoz, M., Calvo-Manzano, J. A., & Corona, B. (2016b). Establishing the State of the Art of Frameworks, Methods and Methodologies Focused on Lightning Software Process: A Systematic Literature Review. *Advances in Intelligent Systems and Computing*, 405, 15–24. http://doi.org/10.1007/978-3-319-26285-7_7

Miramontes, J., Muñoz, M., Calvo-Manzano, J. A., & Corona, B. (2016a). Establecimiento del estado del arte sobre el aligeramiento de procesos de software. *Revista Ibérica de Sistemas Y Tecnologías de Información*. <http://doi.org/10.17013/risti.17.16-25>

Muñoz, M., Mejía, J., Corona, B., Calvo-Manzano, J. A., San Feliu, T., & Miramontes, J. (2016). Analysis of Tools for Assessing the Implementation and Use of Agile Methodologies in SMEs. *Software Process Improvement and Capability Determination*. <http://doi.org/10.1007/978-3-642-30439-2>

Corona, B., Muñoz, M., Calvo-Manzano, J. A., San Feliu, T., & Miramontes, J. (2016). Estado de arte sobre métodos de evaluación de metodologías ágiles en las pymes. *ReCIBE. Revista Electrónica de Computación, Informática, Biomédica Y Electrónica*.

Muñoz, M., Mejía J., Miramontes, J. (2016). Method for Lightning Software Processes through Optimizing the Selection of Software Engineering Best Practices. (Aceptado).

6.3.2. Ponencias en congresos

Ponencia en el 4to Congreso Internacional en Mejora de Procesos de Software (CIMPS 2015), presentando los resultados obtenidos de la revisión sistemática para obtener el estado del arte con respecto al aligeramiento de procesos.

6.3.3. Estancias de Investigación

Estancia de investigación en la Escuela Técnica Superior de Ingenieros Informáticos de la Universidad Politécnica de Madrid, en el departamento de lenguajes y sistemas informáticos

en ingeniería de software. En donde se realizó la obtención del estado del arte con respecto al aligeramiento de procesos de software.

Referencias

- Akbar, R., Hassan, M. F., & Abdullah, A. (2012). A framework of software process tailoring for small and medium size IT companies. *2012 International Conference on Computer & Information Science (ICCIS)*, 2, 914–918. <http://doi.org/10.1109/ICCISci.2012.6297156>
- Al-Tarawneh, M. Y., Abdullah, M. S., & Ali, A. B. M. (2011). A proposed methodology for establishing software process development improvement for small software development firms. *Procedia Computer Science*, 3, 893–897. <http://doi.org/10.1016/j.procs.2010.12.146>
- Barreto, A., Nunes, E., Rocha, A. R., & Murta, L. (2010). Supporting the definition of software processes at consulting organizations via Software Process Lines. *Proceedings - 7th International Conference on the Quality of Information and Communications Technology, QUATIC 2010*, 15–24. <http://doi.org/10.1109/QUATIC.2010.19>
- Calvo-Manzano Villalón, J. A., Cuevas Agustín, G., San Feliu Gilabert, T., De Amescua Seco, A., García Sánchez, L., & Pérez Cota, M. (2002). Experiences in the Application of Software Process Improvement in SMES. *Software Quality Journal*, 261–273.
- CMMI Product Team. (2010). *CMMI® para Desarrollo, Versión 1.3. CMMI Para Desarrollo, Versión 1.3*. Retrieved from <http://www.sei.cmu.edu/library/assets/whitepapers/Spanish Technical Report CMMI V 1 3.pdf>
- Cuevas, G. (2002). Gestión del proceso software. Retrieved from <http://www.agapea.com/libros/Gestion-del-proceso-software-9788480045469-i.htm>
- Daniel Schwabe, G. R. (2008). The Object-Oriented Hypermedia Design Model (OOHDM). Retrieved March 2, 2016, from <http://www.oohdm.telemidia.puc-rio.br/>
- Diaz, J., Garbajosa, J., & Calvo-Manzano, J. a. (2009). Mapping CMMI Level 2 to Scrum Practices: An Experience Report. *Software Process Improvement*, 42, 93–104. <http://doi.org/10.1007/978-3-642-04133-4>
- Django Software Foundation. (2016). Django | The Web framework for perfectionists with deadlines. Retrieved February 29, 2016, from <https://www.djangoproject.com/>
- Ehsan, N., Perwaiz, a., Arif, J., Mirza, E., & Ishaque, a. (2010). CMMI / SPICE based process improvement. *Management of Innovation and Technology (ICMIT), 2010 IEEE International Conference on*, 859–862. <http://doi.org/10.1109/ICMIT.2010.5492803>
- Garzías, J., Pino, F. J., Piattini, M., & Fernández, C. M. (2013). A maturity model for the Spanish software industry based on ISO standards. *Computer Standards and Interfaces*, 35(6), 616–628. <http://doi.org/10.1016/j.csi.2013.04.002>
- Hansson, D. H. (2016). Ruby on Rails. Retrieved February 17, 2016, from <http://rubyonrails.org/>
- IEEE Computer Society. (2014). *Guide to the Software Engineering Body of Knowledge Version 3.0 (SWEBOK Guide V3.0)*.
- ISO. (2004). ISO/IEC 15504-2:2003 - Information technology — Process assessment — Part 2: Performing an assessment. Retrieved from <https://www.iso.org/obp/ui/#iso:std:iso-iec:15504:-2:ed-1:v1:cor:1:v1:en>
- ISO. (2011). ISO/IEC 29110-4-1:2011 Software engineering -- Lifecycle profiles for Very Small Entities (VSEs) -- Part 4-1: Profile specifications: Generic profile group. Retrieved from http://www.iso.org/iso/catalogue_detail.htm?csnumber=51154
- ISO. (2012). ISO/IEC 15504-5:2012 - Information technology -- Process assessment -- Part 5: An

- exemplar software life cycle process assessment model.
- ISO. (2013). ISO/IEC 12207:2008 - Systems and software engineering -- Software life cycle processes. Retrieved from http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=43447
- Kitchenham, B., & Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. *Engineering*, 2, 1051. <http://doi.org/10.1145/1134285.1134500>
- Kruchten, P. (2011). A plea for lean software process models. *Procs. 2011 International Conference on Software and Systems Process (ICSSP)*, 1(604), 235–236. <http://doi.org/10.1145/1987875.1987919>
- Kuilboer, J. ., & Ashrafi, N. (2000). Software process and product improvement: an empirical assessment. *Information and Software Technology*, 42(1), 27–34. [http://doi.org/10.1016/S0950-5849\(99\)00054-3](http://doi.org/10.1016/S0950-5849(99)00054-3)
- Laporte, C., O'Connor, R., Hude, R., & Marvin, J. (2015). ISO/IEC 29110: Systems Engineering Standards for Very Small Enterprises. *Université de Montréal*. Retrieved from <http://substance-en.etsmtl.ca/systems-engineering-iso-29110-standards/>
- Larman, C. (2004). *Agile and Iterative Development: A Manager's Guide*. Addison Wesley.
- Lin, W. L. W., & Fan, X. F. X. (2009). Software Development Practice for FDA-Compliant Medical Devices. *2009 International Joint Conference on Computational Sciences and Optimization*, 2, 388–390. <http://doi.org/10.1109/CSO.2009.191>
- Marçal, A. S. C., De Freitas, B. C. C., Furtado Soares, F. S., & Belchior, A. D. (2007). Mapping CMMI project management process areas to SCRUM practices. *Proceedings - International Conference on Software Engineering*, 13–22. <http://doi.org/10.1109/SEW.2007.102>
- MariaDB Foundation. (2016). MariaDB.org - Continuity and open collaboration. Retrieved March 1, 2016, from <https://mariadb.org/>
- Mundra, A., Misra, S., & Dhawale, C. A. (2013). Practical Scrum-Scrum Team: Way to Produce Successful and Quality Software. *Proceedings of the 2013 13th International Conference on Computational Science and Its Applications, ICCSA 2013*, 119–123. <http://doi.org/10.1109/ICCSA.2013.25>
- NYCE. (2011a). ISO/IEC 29110 – INGENIERIA DE SOFTWARE – PERFILES DEL CICLO DE VIDA EN PEQUEÑAS ENTIDADES. Retrieved November 10, 2015, from http://www.moprosoft.com.mx/contenido.aspx?id_pagina=1118
- NYCE. (2011b). NMX-I-059/02-NYCE-2011 (MoProSoft). Retrieved November 5, 2015, from http://www.moprosoft.com.mx/contenido.aspx?id_pagina=8
- O'Regan, G. (2011). *Introduction to Software Process Improvement*. (I. Mackie, Ed.). Mallow: Springer. <http://doi.org/10.1007/978-0-85729-172-1>
- Oracle Corporation. (2016). MySQL. Retrieved March 1, 2016, from <http://www.mysql.com/>
- Otwell, T. (2016). Laravel - The PHP Framework For Web Artisans. Retrieved February 29, 2016, from <https://laravel.com/>
- Petersen, K., & Wohlin, C. (2010). Software process improvement through the Lean Measurement (SPI-LEAM) method. *Journal of Systems and Software*, 83(7), 1275–1287. <http://doi.org/10.1016/j.jss.2010.02.005>
- Pettersson, F., Ivarsson, M., Gorschek, T., & Öhman, P. (2008). A practitioner's guide to light weight software process assessment and improvement planning. *Journal of Systems and Software*, 81(6), 972–995. <http://doi.org/10.1016/j.jss.2007.08.032>

- Pino, F. J., Baldassare, M. T., Piattini, M., & Visaggio, G. (2009). Harmonizing maturity levels from CMMI-DEV and ISO/IEC 15504. *Journal of Software Maintenance and Evolution*, 18.
- Pino, F. J., Pedreira, O., García, F., Luaces, M. R., & Piattini, M. (2010). Using Scrum to guide the execution of software process improvement in small organizations. *Journal of Systems and Software*, 83(10), 1662–1677. <http://doi.org/10.1016/j.jss.2010.03.077>
- Politecnico di Milano. (2003). The Web Modeling Language. Retrieved March 2, 2016, from <http://www.webml.org/webml/page1.do>
- Ramos, C., & Mendoza, L. (2014). *Implementación del estándar ISO/IEC 29110-4-1 para pequeñas organizaciones de desarrollo de software*. Universidad Peruana de Ciencias Aplicadas - UPC.
- Reifer, D. J. (2000). The CMMI: it's formidable. *Journal of Systems and Software*, 50(2), 97–98. [http://doi.org/10.1016/S0164-1212\(99\)00119-3](http://doi.org/10.1016/S0164-1212(99)00119-3)
- Rios, B. L. F., Vargas, M. A. A., Espinoza, J. M. O., & Peralta, M. D. C. A. (2008). Experiences on the Implementation of MoProSoft and Assessment of Processes under the NMX-I-059/02-NYCE-2005 Standard in a Small Software Development Enterprise. *2008 Mexican International Conference on Computer Science*, 323–328. <http://doi.org/10.1109/ENC.2008.24>
- Rizwan Jameel Qureshi, M., & Hussain, S. a. (2008). An adaptive software development process model. *Advances in Engineering Software*, 39(8), 654–658. <http://doi.org/10.1016/j.advengsoft.2007.08.001>
- Rizwan, M., & Hussain, S. a. (2008). An adaptive software development process model. *Advances in Engineering Software*, 39(8), 654–658. <http://doi.org/10.1016/j.advengsoft.2007.08.001>
- Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2), 131–164. <http://doi.org/10.1007/s10664-008-9102-8>
- Sampieri, R., Fernández, C., & Baptista, P. (2010). *Metodología de la investigación*. Retrieved from <http://www.casadellibro.com/libro-metodologia-de-la-investigacion-5-ed-incluye-cd-rom/9786071502919/1960006>
- Selleri Silva, F., Soares, F. S. F., Peres, A. L., Azevedo, I. M. De, Vasconcelos, A. P. L. F., Kamei, F. K., & Meira, S. R. D. L. (2015). Using CMMI together with agile software development: A systematic review. *Information and Software Technology*, 58, 20–43. <http://doi.org/10.1016/j.infsof.2014.09.012>
- Silva, A. (2004). Modelos de calidad. La industria del software en México. Retrieved November 5, 2015, from <http://www.enterate.unam.mx/Articulos/2004/Enero/modelos.htm>
- The PostgreSQL Global Development Group. (2016). PostgreSQL: The world's most advanced open source database. Retrieved March 1, 2016, from <http://www.postgresql.org/>
- The Standish Group International. (2013). CHAOS MANIFESTO 2013: Think Big, Act Small. *The Standish Group International*, 1–52. Retrieved from <http://www.standishgroup.com>
- Vale, T., Cabral, B., Alvim, L., Soares, L., Santos, A., Machado, I., ... Almeida, E. (2014). SPLICE: A Lightweight Software Product Line Development Process for Small and Medium Size Projects. *2014 Eighth Brazilian Symposium on Software Components, Architectures and Reuse*, 42–52. <http://doi.org/10.1109/SBCARS.2014.11>
- Web Engineering Group. (2015). UWE – UML-based Web Engineering. Retrieved March 2, 2016, from <http://uwe.pst.ifi.lmu.de/index.html>
- Yin, R. K. (2002). *Case study research: design and methods*. *Applied social research method series* (tercera ed, Vol. 5). <http://doi.org/10.1097/FCH.0b013e31822dda9e>

Anexo A. Estudios primarios de la revisión sistemática

- S1 Akbar, R., Hassan, M. F., & Abdullah, A. (2012). A framework of software process tailoring for small and medium size IT companies. *2012 International Conference on Computer & Information Science (ICIS)*, 2, 914–918. <http://doi.org/10.1109/ICISci.2012.6297156>
- S2 Al-Tarawneh, M. Y., Abdullah, M. S., & Ali, A. B. M. (2011). A proposed methodology for establishing software process development improvement for small software development firms. *Procedia Computer Science*, 3, 893–897. <http://doi.org/10.1016/j.procs.2010.12.146>
- S3 Alwardt, A. L., Mikeska, N., Pandorf, R. J., & Tarpley, P. R. (2009). A lean approach to designing for software testability. *AUTOTESTCON (Proceedings)*, 178–183. <http://doi.org/10.1109/AUTEST.2009.5314039>
- S4 Camargo, K. G., Ferrari, F. C., & Fabbri, S. C. (2015). Characterising the state of the practice in software testing through a TMMi-based process. *Journal of Software Engineering Research and Development*, 3(1), 7. <http://doi.org/10.1186/s40411-015-0019-9>
- S5 Edison, H., Wang, X., & Abrahamsson, P. (2015). Lean startup: Why Large Software Companies Should Care. *Scientific Workshop Proceedings of the XP2015 on - XP '15 Workshops*, 1–7. <http://doi.org/10.1145/2764979.2764981>
- S6 Farid, W. M. (2012). The Normap methodology: Lightweight engineering of non-functional requirements for agile processes. *Proceedings - Asia-Pacific Software Engineering Conference, APSEC, 1*, 322–325. <http://doi.org/10.1109/APSEC.2012.23>
- S7 Farrow, A., & Greene, S. (2008). Fast & predictable - A lightweight release framework promotes agility through rhythm and flow. *Proceedings - Agile 2008 Conference*, 224–228. <http://doi.org/10.1109/Agile.2008.83>
- S8 Funkhouser, O., Eitzkorn, L. H., & Hughes, W. E. (2008). A lightweight approach to software validation by comparing UML use cases with internal program documentation selected via call graphs. *Software Quality Journal*, 16(1), 131–156. <http://doi.org/10.1007/s11219-007-9034-3>
- S9 García, J., Amescua, A., Sánchez, M. I., & Bermón, L. (2011). Design guidelines for software processes knowledge repository development. *Information and Software Technology*, 53(8), 834–850. <http://doi.org/10.1016/j.infsof.2011.03.002>
- S10 Ivarsson, M., & Gorschek, T. (2012). Tool support for disseminating and improving development practices. *Software Quality Journal*, 20(1), 173–199. <http://doi.org/10.1007/s11219-011-9139-6>
- S11 Kim, T., Chandra, R., & Zeldovich, N. (2013). Optimizing unit test execution in large software programs using dependency analysis. *Proceedings of the 4th Asia-Pacific Workshop on Systems - APSys '13*, 1–6. <http://doi.org/10.1145/2500727.2500748>
- S12 Kirk, D., & Tempero, E. (2012). A lightweight framework for describing software practices. *Journal of Systems and Software*, 85(3), 582–595. <http://doi.org/10.1016/j.jss.2011.09.024>
- S13 Kruchten, P. (2011). A plea for lean software process models. *Procs. 2011 International Conference on Software and Systems Process (ICSSP)*, 1(604), 235–236. <http://doi.org/10.1145/1987875.1987919>
- S14 Lehtinen, T. O. a, Mäntylä, M. V., & Vanhanen, J. (2011). Development and evaluation of a lightweight root cause analysis method (ARCA method) - Field studies at four software companies. *Information and Software Technology*, 53(10), 1045–1061. <http://doi.org/10.1016/j.infsof.2011.05.005>
- S15 Lin, W. L. W., & Fan, X. F. X. (2009). Software Development Practice for FDA-Compliant Medical Devices. *2009 International Joint Conference on Computational Sciences and Optimization*, 2, 388–390. <http://doi.org/10.1109/CSO.2009.191>
- S16 Misra, S. C., Kumar, V., & Kumar, U. (2009). Identifying some important success factors in adopting agile software development practices. *Journal of Systems and Software*, 82(11), 1869–1890. <http://doi.org/10.1016/j.jss.2009.05.052>
- S17 Motta, A., & Mangano, N. (2013). Lightweight Analysis of Software Design Models at the Whiteboard, 18–23.
- S18 Pang, H., Zhou, L., & Chang, X. (2011). Lightweight web framework oriented on page flow component. *Proceedings 2011 International Conference on Mechatronic Science, Electric Engineering and Computer, MEC 2011*, 1248–1251. <http://doi.org/10.1109/MEC.2011.6025694>
- S19 Park, S., & Bae, D.-H. (2011). An approach to analyzing the software process change impact using process slicing and simulation. *Journal of Systems and Software*, 84(4), 528–543. <http://doi.org/10.1016/j.jss.2010.11.919>
- S20 Peng, X., Chen, B., Yu, Y., & Zhao, W. (2012). Self-tuning of software systems through dynamic quality tradeoff and value-based feedback control loop. *Journal of Systems and Software*, 85(12), 2707–2719. <http://doi.org/10.1016/j.jss.2012.04.079>
- S21 Petersen, K., & Wohlin, C. (2010). Software process improvement through the Lean Measurement (SPI-LEAM) method. *Journal of Systems and Software*, 83(7), 1275–1287. <http://doi.org/10.1016/j.jss.2010.02.005>
- S22 Pettersson, F., Ivarsson, M., Gorschek, T., & Öhman, P. (2008). A practitioner's guide to light weight software process assessment and improvement planning. *Journal of Systems and Software*, 81(6), 972–995. <http://doi.org/10.1016/j.jss.2007.08.032>
- S23 Pino, F. J., Pedreira, O., García, F., Luaces, M. R., & Piattini, M. (2010). Using Scrum to guide the execution of software process improvement in small organizations. *Journal of Systems and Software*, 83(10), 1662–1677. <http://doi.org/10.1016/j.jss.2010.03.077>
- S24 Rapp, D., Hess, A., Seyff, N., Spörri, P., Fuchs, E., & Glinz, M. (2014). Lightweight Requirements Engineering Assessments in Software Projects, 354–363.
- S25 Rigby, P., Cleary, B., Painchaud, F., Storey, M. A., & German, D. (2012). Contemporary peer review in action: Lessons from open source development. *IEEE Software*, 29(6), 56–61. <http://doi.org/10.1109/MS.2012.24>
- S26 Rodríguez, P., Mikkonen, K., Kuvaja, P., Oivo, M., & Garbajosa, J. (2013). Building lean thinking in a telecom software development organization: strengths and challenges. *Proceedings of the 2013 International Conference on Software and System Process - ICSSP 2013*, 98. <http://doi.org/10.1145/2486046.2486064>
- S27 Rubin, E., & Rubin, H. (2011). Supporting agile software development through active documentation. *Requirements Engineering*, 16(2), 117–132. <http://doi.org/10.1007/s00766-010-0113-9>
- S28 Selleri, F., Santana, F., Soares, F., Lima, A., Monteiro, I., Azevedo, D., ... Meira, D. L. (2015). Using CMMI together with agile software development: A systematic review. *Information and Software Technology*, 58, 20–43. <http://doi.org/10.1016/j.infsof.2014.09.012>
- S29 Stankovic, D., Nikolic, V., Djordjevic, M., & Cao, D.-B. (2013). A survey study of critical success factors in agile software projects in former Yugoslavia IT companies. *Journal of Systems and Software*, 86(6), 1663–1678. <http://doi.org/10.1016/j.jss.2013.02.027>
- S30 Vale, T., Cabral, B., Alvim, L., Soares, L., Santos, A., Machado, I., ... Almeida, E. (2014). SPLICE: A Lightweight Software Product Line Development Process for Small and Medium Size Projects. *2014 Eighth Brazilian Symposium on Software Components, Architectures and Reuse*, 42–52. <http://doi.org/10.1109/SBCARS.2014.11>
- S31 Vanhanen, J., Mäntylä, M. V., & Itkonen, J. (2009). Lightweight elicitation and analysis of software product quality goals - A multiple industrial case study. *2009 3rd International Workshop on Software Product Management, IWSPM 2009*, 27–30. <http://doi.org/10.1109/IWSPM.2009.5>
- S32 Zarour, M., Abran, A., Desharnais, J.-M., & Alarifi, A. (2015). An investigation into the best practices for the successful design and implementation of lightweight software process assessment methods: A systematic literature review. *Journal of Systems and Software*, 101, 180–192. <http://doi.org/10.1016/j.jss.2014.11.041>

Anexo B. Tipo de propuestas y objetivos de los estudios primarios

ID	Tipo de propuesta	Objetivo resumido
S1	Marco de trabajo	Presentar un marco de trabajo meta-modelo ligero para la adaptación de procesos de software.
S2	Metodología	Propuesta de una metodología para la mejora de procesos en PyMES utilizando CMMI
S3	Marco de trabajo	Mostrar como Lean 123 con enfoque automático en pruebas de software ahorra costes y mejora la calidad de los productos.
S4	Herramienta	Identificar un conjunto de prácticas clave para apoyar un proceso genérico ligero para pruebas de software basado en TMMi (Test Maturity Model integration).
S5	Método	Investigar por qué las grandes empresas deben adoptar Lean Startup para buscar la innovación radical.
S6	Metodología	Proponer NORMAP, una metodología ligera para requerimientos no funcionales en procesos ágiles.
S7	Marco de trabajo	Presentar un marco de trabajo ligero para la liberación de software, con el fin de optimizar en tiempo la entrega de alta calidad a los clientes.
S8	Metodología	Presentar una metodología para automatizar la validación de software.
S9	Herramienta	Diseñar guías para la implementación de una biblioteca de activos de procesos mediante una wiki, basada en Process Asset Library (PAL) para almacenar las mejores prácticas.
S10	Herramienta	Presentar una herramienta de apoyo con un enfoque Experience Factory para difundir y mejorar las prácticas utilizadas en la organización.
S11	Herramienta	Demostrar que la herramienta TAO puede reducir las pruebas de unidad en tiempo de ejecución en más del 96%, en dos grandes proyectos de software.
S12	Marco de trabajo	Desarrollar un marco de trabajo para capturar las mejores prácticas de empresas exitosas.
S13	Otro*	Explicar porque los procesos grandes y pesados no son adecuados para el desarrollo de software y se debe optar por procesos ligeros.
S14	Método	Presentar un método ligero para el RCA (Root Cause Analysis) llamado ARCA, en el cual la detección de un problema se basa en la reunión de un grupo enfocado al problema.
S15	Metodología	Mostrar la práctica al desarrollar software para dispositivos médicos que necesitan procesos formales y rigurosos mediante una metodología híbrida entre CMMI desarrollo ágil de software.
S16	Marco de trabajo	Presentar el estado del arte sobre la identificación de factores que influyen en el éxito de los proyectos desde la perspectiva de profesionales en desarrollo ágil de software.
S17	Herramienta	Mejorar la herramienta Calico para que permita un análisis ligero dándole retroalimentación al desarrollador mientras esboza un diseño.
S18	Marco de trabajo	Mostrar el marco de trabajo ligero PageFlow que tiene el objetivo de mejorar el desarrollo de software y reducir los costos de desarrollo.
S19	Herramienta	Propuesta de un enfoque para analizar el impacto del cambio de un proceso de software de una manera ligera usando slicing y simulation.
S20	Método	Proponer un método de autoajuste dinámicamente pueda captar los requisitos de calidad y tomar decisiones sobre la compensación (trade-off) de atributos de caldead.
S21	Método	Proponer un enfoque para reunir el paradigma para la mejora de procesos de Software a través de la medición Lean (SPI-LEAM).
S22	Marco de trabajo	Presentar una guía para la evaluación ligera de procesos y planificación de mejora utilizando el marco de trabajo iFLAP.
S23	Marco de	Propuesta de un “proceso ligero para incorporar mejoras”, utilizando la filosofía del método ágil Scrum, con el

	trabajo	objetivo de dar guías detalladas para incorporar mejoras de procesos en pequeñas empresas.
S24	Método	Desarrollar un método ligero para responder preguntas relacionadas con la calidad de los procesos de Ingeniería de Requisitos, para que una empresa pueda evaluarse y mejorar.
S25	Herramienta	Describir lecciones de los procesos de revisión de código en OSS (Open Source Software) para transferirlos al desarrollo de software propietario.
S26	Metodología	Explorar como los principio Lean son implementados en empresas de desarrollo de software y los retos en la implementación de Lean.
S27	Marco de trabajo	Proponer el diseño del sistema ADSD (Active Documentation Software Design), de manera que el código fuente incorpore las sentencias de documentación.
S28	Otro*	Evaluar, sintetizar y presentar resultados en el uso de CMMI en combinación con desarrollo ágil de software, y después de eso dar una visión general de los temas investigados.
S29	Otro*	Presentar los resultados de un estudio empírico para determinar los factores críticos que influyen en el éxito de los proyectos ágiles.
S30	Método	Presentar a SPLICE, un proceso de desarrollo ligero que combina prácticas ágiles con líneas de productos de software.
S31	Método	Presentar un método ligero para la obtención y análisis de objetivos de calidad de productos de software.
S32	Otro *	Presentar resultados de una revisión sistemática de la literatura donde se obtienen las mejores prácticas para hacer una evaluación de procesos de software ligera.

* En estos estudios el tema del aligeramiento de procesos de software es analizado, pero no se presenta una propuesta.

Anexo C. Cuestionario de prácticas específicas de Planificación de Proyectos

Objetivo específico 1: Establecer estimaciones

1.1. ¿Se estima el alcance del proyecto?

Ejemplo: Establecer un WBS (Estructura de descomposición de trabajo) para estimar el alcance del proyecto.

1.2. ¿Se establecen las estimaciones de los atributos de los productos de trabajo y de las tareas?

Ejemplos de atributos para estimar son:

- Número y complejidad de los requisitos.
- Número y complejidad de las interfaces.
- Número de funciones.
- Puntos función.
- Líneas de código fuente.
- Número de clases y de objetos.

1.3. ¿Se definen las fases del ciclo de vida del proyecto?

Ejemplo de productos de trabajo:

- Fases del ciclo de vida del proyecto.

1.4. ¿Se estima el esfuerzo y coste del proyecto?

Ejemplos de productos de trabajo

- Análisis razonado de la estimación.
- Estimaciones del esfuerzo del proyecto.
- Estimaciones del coste del proyecto.

Objetivo específico 2: Establecer estimaciones

2.1. ¿Se establece el presupuesto y el calendario del proyecto?

Ejemplo de productos de trabajo:

- Calendarios del proyecto
- Dependencias del calendario
- Presupuesto del proyecto

2.2. ¿Se identifican los riesgos del proyecto?

Ejemplo de productos de trabajo:

- Riesgos identificados
- Impactos y probabilidad de ocurrencia de los riesgos
- Prioridades de los riesgo

2.3. ¿Se planifica la gestión de los datos del proyecto?

Ejemplo de productos de trabajo:

- Plan para la gestión de datos
- Lista maestra de datos gestionados
- Calendario para la recogida de datos del proyecto
- Listado de datos del proyecto a recoger

2.4. ¿Se planifican los recursos del proyecto?

Ejemplo de productos de trabajo:

- Paquetes de trabajo.
- Diccionario de tareas de la WBS.
- Requisitos de personal basados en el tamaño y el alcance del proyecto.
- Lista de instalaciones y equipamientos críticos.
- Lista de requisitos de administración del proyecto.

2.5. ¿Se planifica el conocimiento y las habilidades necesarias?

Ejemplo de productos de trabajo:

- Inventario de habilidades necesarias.
- Planes de personal y nuevas contrataciones.
- Bases de datos (p. ej., habilidades, formación).
- Planes de formación.

2.6. ¿Se planifica la involucración de las partes interesadas?

Ejemplo de productos de trabajo:

- Plan para la involucración de las partes interesadas

2.7. ¿Se establece el plan del proyecto?

Ejemplo de productos de trabajo:

- Plan global del proyecto

Objetivo específico 3: Obtener un compromiso con el plan

3.1. ¿Se revisan los planes que afectan al proyecto?

Ejemplo de productos de trabajo:

- Registro de la revisiones de los planes que afectan al proyecto.

3.2. ¿Se concilian los niveles de trabajo y de recursos?

Ejemplo de productos de trabajo:

- Métodos y parámetros de estimación correspondientes modificados (p. ej., mejores herramientas, uso de productos comerciales).
- Presupuestos renegociados.
- Calendarios modificados.

- Lista de requisitos modificada.

1.5. ¿Se obtiene un compromiso con el plan?

Ejemplo de productos de trabajo:

- Peticiones de compromisos documentadas.
- Compromisos documentados

Anexo D. Mapeo de prácticas del área de Planificación de Proyectos a prácticas de Scrum

Fuentes:

- Mapping CMMI Project Management Process Areas to SCRUM Practices (Marçal et al., 2007)
- Mapping CMMI Level 2 to Scrum Practices: An Experience Report (Diaz et al., 2009).

Tabla 29. Criterios para definir el estado del mapeo de prácticas

Estado	Criterio
<input type="radio"/> No soportada	La práctica no se aborda en Scrum.
<input checked="" type="radio"/> Parcialmente soportada	Hay alguna evidencia de que la práctica es abordada por Scrum, sin embargo la práctica no se aborda completamente.
<input type="radio"/> Soportada	La práctica se aborda completamente.

Objetivo específico 1: Establecer estimaciones

Tabla 30. Mapeo de prácticas específicas a prácticas de Scrum (Objetivo específico 1)

Estado	Práctica específica	Práctica de Scrum						
<input type="radio"/>	<p>1.1. Estimar el alcance del proyecto Ejemplo: Establecer un WBS (Estructura de descomposición de trabajo) para estimar el alcance del proyecto.</p>	<p>No soportada En Scrum el WBS se compone de Product Backlog y de los Sprints predefinidos, sin embargo, el Product Backlog es elaborado por el cliente y presentado por el Product Owner, sin presentar guías sobre cómo se éste se crea.</p>						
<input checked="" type="radio"/>	<p>1.2. Establecer las estimaciones de los atributos de los productos de trabajo y de las tareas. Ejemplos de atributos para estimar son:</p> <ul style="list-style-type: none"> • Número y complejidad de los requisitos. • Número y complejidad de las interfaces. • Número de funciones. • Puntos función. 	<p>Estimar atributos mediante Planning Poker o Magic Estimation Scrum establece una primera estimación en la fase pre-game y una estimación iterativa al principio del sprint (reunión de planificación). Las estimaciones se basan generalmente en tamaño o complejidad atributos.</p> <p>Guías para su implementación (Estimation meeting)</p> <table border="1"> <thead> <tr> <th>Ingredientes:</th> <th>Procedimiento:</th> <th>Duración:</th> </tr> </thead> <tbody> <tr> <td> <ul style="list-style-type: none"> • El Product Backlog, el cual es priorizado por el Product Owner de acuerdo con el valor del negocio. • Tarjetas numeradas • Planing Poker o </td> <td> <ol style="list-style-type: none"> 1. El Product Owner presenta los ítems del Product Backlog que necesitan se estimados por el Equipo. 2. El equipo utiliza Planning Poker or Magic Estimation para estimar los ítems del Backlog. 3. En caso de que un ítem sea muy largo, el equipo lo dividirá en ítems más pequeños. </td> <td> <p>Fijar el tiempo a 35 minutos o menos.</p> <p>Técnicas: - Planning Poker</p> </td> </tr> </tbody> </table>	Ingredientes:	Procedimiento:	Duración:	<ul style="list-style-type: none"> • El Product Backlog, el cual es priorizado por el Product Owner de acuerdo con el valor del negocio. • Tarjetas numeradas • Planing Poker o 	<ol style="list-style-type: none"> 1. El Product Owner presenta los ítems del Product Backlog que necesitan se estimados por el Equipo. 2. El equipo utiliza Planning Poker or Magic Estimation para estimar los ítems del Backlog. 3. En caso de que un ítem sea muy largo, el equipo lo dividirá en ítems más pequeños. 	<p>Fijar el tiempo a 35 minutos o menos.</p> <p>Técnicas: - Planning Poker</p>
Ingredientes:	Procedimiento:	Duración:						
<ul style="list-style-type: none"> • El Product Backlog, el cual es priorizado por el Product Owner de acuerdo con el valor del negocio. • Tarjetas numeradas • Planing Poker o 	<ol style="list-style-type: none"> 1. El Product Owner presenta los ítems del Product Backlog que necesitan se estimados por el Equipo. 2. El equipo utiliza Planning Poker or Magic Estimation para estimar los ítems del Backlog. 3. En caso de que un ítem sea muy largo, el equipo lo dividirá en ítems más pequeños. 	<p>Fijar el tiempo a 35 minutos o menos.</p> <p>Técnicas: - Planning Poker</p>						

		Magic Estimation.	4. Vuelva a calcular todos los ítems del Backlog que puedan ser jalados durante los siguientes tres Sprints. 5. Identifica los ítems del Backlog que necesitan ser aclarados por el Product Owner hasta la siguiente reunión de estimación.	- Magic Estimation
	1.3. Definir las fases del ciclo de vida del proyecto.	Utilizar el ciclo de vida de Scrum		
	Ejemplos de productos de trabajo:	El ciclo de vida de Scrum se compone de 4 fases (Larman, 2004):		
	<ul style="list-style-type: none"> Fases del ciclo de vida del proyecto. 	<ul style="list-style-type: none"> Planning: Establece una visión del proyecto y las expectativas de los interesados, más allá de asegurar la financiación / presupuesto para la ejecución del proyecto. Staging: Identifica y prioriza los requisitos (al menos, para el siguiente sprint). Divide el Product Backlog en Sprints de acuerdo con la priorización anterior, teniendo en cuenta la productividad del equipo. Development: Implementa el sistema en un conjunto de iteraciones de 30 días (Sprints), cuando, al final de cada sprint, un incremento del producto se presenta a las partes interesadas. Release: Despliegue del sistema. 		
	1.4. Estimar el esfuerzo y coste	Estimar el esfuerzo de las historias de usuario		
	Ejemplos de productos de trabajo:	Los practicantes de Scrum estiman el esfuerzo de las historias de usuario en días basándose en Sprints previos, proyectos anteriores y la complejidad relativa de las historias de usuario.		
	<ul style="list-style-type: none"> Análisis razonado de la estimación. Estimaciones del esfuerzo del proyecto. Estimaciones del coste del proyecto. 	El costo no se menciona explícitamente en SCRUM, por lo cual esta práctica es parcialmente satisfecha.		
		Guías para su implementación (Sprint Planning Meeting #1)		
		Ingredientes:	Procedimiento:	Duración:
		<ul style="list-style-type: none"> El Product Backlog estimado y priorizado. Rotafolios, marcadores, tijeras, notas adhesivas, pizarras, lápices, etc. 	<ol style="list-style-type: none"> Comenzar con el primer ítem del Product Backlog (Historia). El equipo discute y analiza los requerimientos de la historia. El equipo elabora las pruebas de aceptación de usuario. Anotar las restricciones. Definir los criterios de aceptación. Determinar el nivel de Hecho para la historia. Dibujar bocetos de las funcionalidades de la historia. Para el siguiente ítem del Backlog, regresar al paso 1. 	60 minutos por semana de Sprint.
				Herramientas:
				- Burn down chart - Burn up chart

Objetivo específico 2: Desarrollar un plan de proyecto

Tabla 31. Mapeo de prácticas específicas a prácticas de Scrum (Objetivo específico 2)

Estado	Práctica específica	Práctica de Scrum
	2.1. Establecer el presupuesto y el calendario	Establecer presupuesto y sprints (calendario)
	Ejemplos de productos de trabajo:	Durante la fase pre-game se establecen hitos (objetivos del sprint) y calendarios (sprints) de acuerdo al Backlog inicial. Presupuesto o hitos adicionales pueden ser asignados al proyecto en cada sprint durante su planificación.
	<ul style="list-style-type: none"> Calendarios del proyecto Dependencias del calendario Presupuesto del proyecto 	Por otra parte, Scrum no provee orientaciones sobre el establecimiento de presupuesto, por lo cual esta práctica es parcialmente satisfecha.
		Guías para su implementación (Sprint planning meeting 2)
		Ingredientes:
		<ul style="list-style-type: none"> El Product Backlog seleccionado para el Sprint. Rotafolios,
		Procedimiento:
		<ol style="list-style-type: none"> Comenzar con el primer ítem del Product Backlog. Recapitular la comprensión del equipo de lo que es realmente deseado examinando los rotafolios de reunión de planificación de Sprint # 1. Ejecutar una discusión orientada a como podría ser

	<p>marcadores, tijeras, notas adhesivas, pizarras, lápices, etc.</p>	<p>implementado el ítem del Backlog. Estas preguntas pueden ayudar:</p> <ol style="list-style-type: none"> ¿Qué interfaces necesitamos escribir? ¿Qué bases de datos necesitaremos? ¿Qué arquitectura es necesaria? <p>Duración: 60 minutos por semana de Sprint.</p> <p>Una vez que el equipo tiene un claro entendimiento de la manera en la que desea construir una característica, se puede pasar al siguiente ítem del Backlog.</p>
<p>2.2. Identificar los riesgos del proyecto</p> <p>Ejemplos de productos de trabajo:</p> <ul style="list-style-type: none"> • Riesgos identificados • Impactos y probabilidad de ocurrencia de los riesgos • Prioridades de los riesgo 	<p>Identificar lista de impedimentos</p> <p>SCRUM considera un riesgo como un posible impedimento para el proyecto. La identificación de los riesgos se produce de forma iterativa, durante las reuniones diarias registrando en pizarras, rotafolios o la lista de impedimentos.</p> <p>Sin embargo, la identificación de riesgos no se produce de una manera sistemática utilizando las categorías de riesgo y sus fuentes. Por lo tanto, esta práctica es parcialmente satisfecha.</p>	<p>Guías para su implementación (Daily Scrum)</p> <p>Ingredientes:</p> <ul style="list-style-type: none"> • Taskboard (Tabla de tareas) • Notas adhesivas • Marcadores <p>Duración: 15 minutos, a la misma hora y ubicación todos los días.</p> <p>Procedimiento:</p> <ol style="list-style-type: none"> 1. El equipo se reúne alrededor del Taskboard. 2. Uno de los miembros del equipo comienza explicando a sus compañeros lo que ha logrado poner en práctica desde el último Scrum Diario. 3. El miembro del equipo mueve la tarea correspondiente en la columna correcta de la TaskBoard. 4. En su caso, el miembro del Equipo recoge una nueva tarea y la coloca en la columna "Work in Progress". <p>Si el miembro del equipo ha detectado un problema o impedimento desde el último Scrum Diario, lo reporta al ScrumMaster quien toma nota para la eliminación inmediata.</p>
<p>2.3. Planificar la gestión de los datos</p> <p>Ejemplos de productos de trabajo:</p> <ul style="list-style-type: none"> • Plan para la gestión de datos • Lista maestra de datos gestionados • Calendario para la recogida de datos del proyecto • Listado de datos del proyecto a recoger 	<p>No soportada</p> <p>En Scrum cualquier dato generado en el proyecto se almacenan en carpetas públicas o pizarras al alcance de todos, pero no hay un plan de gestión de datos formal o procedimiento para recoger estos datos.</p>	
<p>2.4. Planificar los recursos del proyecto</p> <p>Ejemplos de productos de trabajo:</p> <ul style="list-style-type: none"> • Paquetes de trabajo. • Diccionario de tareas de la WBS. • Requisitos de personal basados en el tamaño y el alcance del proyecto. • Lista de requisitos de administración del proyecto. 	<p>Establecer el equipo Scrum</p> <p>Durante la fase pre-game, los requerimientos de personal y la lista de equipamiento se definen. Como resultado se establece el Equipo Scrum. Durante la ejecución sprints, el ScrumMaster es el encargado de proporcionar nuevos recursos que sean necesarios.</p>	<p>Guías para su implementación (Sprint Planning Meeting #2)</p> <p>Ingredientes:</p> <ul style="list-style-type: none"> • El Product Backlog seleccionado para el Sprint. • Rotafolios, marcadores, tijeras, notas adhesivas, pizarras, lápices, etc. <p>Procedimiento:</p> <ol style="list-style-type: none"> 1. Comenzar con el primer ítem del Product Backlog. 2. Recapitular la comprensión del equipo de lo que es realmente deseado examinado los rotafolios de reunión de planificación de Sprint # 1. 3. Ejecutar una discusión orientada a como podría ser implementado el ítem del Backlog. Estas preguntas pueden ayudar: <ol style="list-style-type: none"> a. ¿Qué interfaces necesitamos escribir? b. ¿Qué bases de datos necesitaremos? c. ¿Qué arquitectura es necesaria?

	Duración: 60 minutos por semana de Sprint.	Una vez que el equipo tiene un claro entendimiento de la manera en la que desea construir una característica, se puede pasar al siguiente ítem del Backlog.
○	2.5. Planificar el conocimiento y las habilidades necesarias Ejemplos de productos de trabajo: • Inventario de habilidades necesarias. • Planes de personal y nuevas contrataciones. • Bases de datos (p. ej., habilidades, formación). • Planes de formación.	No soportada En Scrum los equipos son multifuncionales y auto-gestionados, el conocimiento y las habilidades necesarias se identifican durante la fase pre-game. Sin embargo, Scrum no menciona la necesidad de planificar el conocimiento y habilidades necesarias para realizar las actividades del proyecto.
●	2.6. Planificar la involucración de las partes interesadas Ejemplos de productos de trabajo: • Plan para la involucración de las partes interesadas.	Definir roles y responsabilidades al inicio y fin del Sprint Scrum define roles, responsabilidades y participación de las partes interesadas al principio y al final de cada sprint. Esta participación es supervisada por el Scrum Master y registrada en un plan de comunicación.
●	2.7. Establecer el plan del proyecto Ejemplos de productos de trabajo: • Plan global del proyecto.	Establecer el plan del proyecto con base a la visión y el Product Backlog del proyecto El documento de visión y el Producto Backlog crean una base para la elaboración de un plan de proyecto de alto nivel. La visión describe por qué el proyecto se está realizando y lo cual es el estado final deseado. El Producto Backlog define los requisitos funcionales y no funcionales que el sistema debe cumplir para entregar la visión, priorizados y no estimados. Guías para su implementación (Taskboard): Herramientas: -Taskboard: Ofrece una descripción visual de los ítems seleccionados del Product Backlog (historias) y el Sprint Backlog (tareas). a) El taskboard sólo es mantenido por el equipo. b) Nada vence la experiencia de usar un tablero físico en la pared o sala del equipo. c) Si hay equipos de software distribuidos, un software para el taskboard le ayudará a mantener la comunicación entre distancias. Procedimiento: Crear un tablero con las siguientes columnas: 1. Selected Product Backlog (Historias). Esta columna contiene todos los ítems del Backlog que el equipo desea implementar en el Sprint actual, en orden priorizado. 2. Task To Do. Son el resultado del Sprint Planning Meeting #2 o se han agregado durante el Sprint. 3. Work in Progress. Cuando un miembro del equipo inicia una tarea, se mueve la nota adhesiva de esta tarea a esta columna. 4. Done. Cuando se ha completado una tarea, se mueve la nota adhesiva a esta columna.

Objetivo específico 3: Obtener un compromiso con el plan

Tabla 32. Mapeo de prácticas específicas a prácticas de Scrum (Objetivo específico 2)

Estado	Práctica específica	Práctica de Scrum
●	3.1. Revisar los planes que afectan al proyecto Ejemplos de productos de trabajo: • Registro de las revisiones que afectan al proyecto	Revisar los planes que afectan al proyecto durante las reuniones de planificación y retrospectivas En Scrum, los planes se revisan al comienzo de cada sprint y las posibles adaptaciones se llevan a cabo conforme al cambio de requisitos y tecnologías. Guías para su implementación (Sprint review) Ingredientes: - Incremento del producto potencialmente entregable, presentado por el Procedimiento: 1. El Product Owner da la bienvenida a los participantes del Sprint Review 2. El Product Owner les recuerda a los presentes el propósito del Sprint pasado: el objetivo del Sprint, el cual soporta las historias que el equipo había

	<p>equipo.</p> <ul style="list-style-type: none"> - Taskboard, notas adhesivas, marcadores. <p>Duración: 90 min, al final del Sprint.</p>	<p>seleccionado.</p> <ol style="list-style-type: none"> 3. Los miembros del equipo de desarrollo demuestran las nuevas funcionalidades y piden al usuario final que las prueben. 4. El Scrum Master modera la sesión. 5. La retroalimentación del usuario final es documentada por el ScrumMaster.
<p>3.2. Conciliar los niveles de trabajo y de recursos</p> <p>Ejemplo de productos de trabajo:</p> <ul style="list-style-type: none"> • Métodos y parámetros de estimación correspondientes modificados (p. ej., mejores herramientas, uso de productos comerciales). • Presupuestos renegociados. • Calendarios modificados. • Lista de requisitos modificada. 	<p>Conciliar los niveles de trabajo y recursos durante la planificación de Sprints</p> <p>Durante la reunión de planificación de Sprint se produce la conciliación de trabajo debido a que el Backlog es dinámico, por lo que son posibles nuevas estimaciones y calendarios.</p> <p>El equipo, el Product Owner y el Scrum Master definen las funcionalidades que se desarrollarán en el Sprint.</p> <p>Guías para su implementación (Sprint Planning Meeting #2)</p> <p>Ingredientes:</p> <ul style="list-style-type: none"> • El Product Backlog seleccionado para el Sprint. • Rotafolios, marcadores, tijeras, notas adhesivas, pizarras, lápices, etc. <p>Duración: 60 minutos.</p>	<p>Procedimiento:</p> <ol style="list-style-type: none"> 1. Comenzar con el primer ítem del Product Backlog. 2. Recapitular la comprensión del equipo de lo que es realmente deseado examinando los rotafolios de reunión de planificación de Sprint # 1. 3. Ejecutar una discusión orientada a como podría ser implementado el ítem del Backlog. Estas preguntas pueden ayudar: <ol style="list-style-type: none"> a. ¿Qué interfaces necesitamos escribir? b. ¿Qué bases de datos necesitaremos? c. ¿Qué arquitectura es necesaria? <p>Una vez que el equipo tiene un claro entendimiento de la manera en la que desea construir una característica, se puede pasar al siguiente ítem del Backlog.</p>
<p>3.3. Obtener el compromiso con el plan</p> <p>Ejemplo de productos de trabajo:</p> <ul style="list-style-type: none"> • Peticiones de compromisos documentadas. • Compromisos documentados. 	<p>Obtener el compromiso con el plan durante la planificación de Sprints</p> <p>El compromiso del plan se produce de forma continua al comienzo de cada sprint, durante la reunión de planificación de Sprint. En ella el ScrumMaster pregunta al equipo si desea comprometerse con cada ítem del Product Backlog.</p> <p>Guías para su implementación (Sprint Planning Meeting #1)</p> <p>Ingredientes:</p> <ul style="list-style-type: none"> • El Product Backlog estimado y priorizado. • Rotafolios, marcadores, tijeras, notas adhesivas, pizarras, lápices, etc. 	<p>Procedimiento:</p> <ol style="list-style-type: none"> 1. Comenzar con el primer ítem del Product Backlog (Historia). 2. El equipo discute y analiza los requerimientos de la historia. 3. El equipo elabora las pruebas de aceptación de usuario. 4. Anotar las restricciones. 5. Definir los criterios de aceptación. 6. Determinar el nivel de Hecho para la historia. 7. Dibujar bocetos de las funcionalidades de la historia. 8. Para el siguiente ítem del Backlog. 9. Continuar en el paso 2. <p>Una vez elegidos los ítems que se van a desarrollar en el Sprint, el ScrumMaster le pregunta al equipo si se compromete a entregar dichos ítems.</p>

Anexo E. Autorización de publicación en formato electrónico de tesis



CENTRO DE INVESTIGACIÓN EN MATEMÁTICAS, A.C.

BIBLIOTECA

AUTORIZACION PUBLICACION EN FORMATO ELECTRONICO DE TESIS

El que suscribe
Autor(s) de la tesis: Juan José Miramontes Sandoval

Título de la tesis: Método para aligerar procesos de software mediante la optimización en la selección de prácticas de Ingeniería de Software.

Institución y Lugar: Centro de Investigación en Matemáticas. Zacatecas, Zacatecas.

Grado Académico: Licenciatura () Maestría (X) Doctorado () Otro () -----
2016

Año de presentación: Ingeniería de Software

Área de Especialidad: Dra. Mirna Ariadna Muñoz Mata, Dr. Jezreel Mejía Miranda

Director(es) de Tesis: juan.miramontes@cimat.mx

Correo electrónico: Software process lightening, Software process improvement, Added value analysis, agile development practices.

Domicilio: agile development practices.

Palabra(s) Clave(s):

Por medio del presente documento autorizo en forma gratuita a que la Tesis arriba citada sea divulgada y reproducida para publicarla mediante almacenamiento electrónico que permita acceso al público a leerla y conocerla visualmente, así como a comunicarla públicamente en la Página WEB del CIMAT.

La vigencia de la presente autorización es por un periodo de 3 años a partir de la firma de presente instrumento, quedando en el entendido de que dicho plazo podrá prorrogar automáticamente por periodos iguales, si durante dicho tiempo no se revoca la autorización por escrito con acuse de recibo de parte de alguna autoridad del CIMAT

La única contraprestación que condiciona la presente autorización es la del reconocimiento del nombre del autor en la publicación que se haga de la misma.

Atentamente

Juan José Miramontes Sandoval

Nombre y firma del tesista

CALLE JALISCO S/N MINERAL DE VALENCIANA APDO. POSTAL 402
C.P. 36240 GUANAJUATO, GTO., MÉXICO
TELÉFONOS (473) 732 7155, (473) 735 0800 EXT. 49609 FAX. (473) 732 5749
E-mail: biblioteca@cimat.mx