

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA EN SISTEMAS

**PROPUESTA DE MEJORAS PARA EQUIPOS DE PRUEBAS EN BASE AL ANÁLISIS
DE INCIDENCIAS NEGATIVAS REPORTADAS POR CLIENTES**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL GRADO DE MAGISTER EN SOFTWARE MENCIÓN
CALIDAD**

Paúl Gerardo Pulgar Salazar
iopaul@hotmail.com

Director: PhD. Rosa del Carmen Navarrete Rueda
rosa.navarrete@epn.edu.ec

Septiembre 2019

CERTIFICACIÓN

Certifico que el presente trabajo de investigación fue conducido por Paúl Gerardo Pulgar Salazar bajo mi supervisión.

PhD. Rosa del Carmen Navarrete Rueda
Director

DECLARACIÓN

Yo, Paúl Gerardo Pulgar Salazar, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Paúl Gerardo Pulgar Salazar

DEDICATORIA

A mi amada esposa Carmen, y a mis hijos Tomás y Sofía.

Paúl

AGRADECIMIENTO

A Dios, por siempre estar a mi lado.

A mi esposa, por brindarme el apoyo sin el cual este trabajo simplemente no habría sido posible. Cada palabra de aliento la llevo conmigo.

A mi padre, que me enseñó a buscar mi propio camino, y que debo confiar en los hechos más que en las palabras.

A mi familia, que me ayudó a continuar y ver la meta en lugar de ver solo los problemas en el camino.

A mi directora de tesis PhD. Rosa del Carmen Navarrete Rueda, por su guía y su confianza en mi. El tiempo dedicado a este trabajo y sus recomendaciones me han permitido mejorar en cada paso.

A mis compañeros de trabajo por sus ideas y consejos. Por toda su ayuda.

A la empresa en la que trabajo, por brindarme todo el apoyo posible.

Finalmente, a todos los que contribuyeron para que pueda conseguir una meta más en mi vida.

Paúl Pulgar

Resumen

El presente estudio presenta una alternativa que puede ser aplicada para el mejoramiento del área de pruebas de software dentro de empresas dedicadas a proveer software como servicio (SaaS). En la actualidad, las aplicaciones que se ofrecen son críticas para el buen funcionamiento de las empresas; por lo que, si presentan problemas, causarían que las empresas no puedan cumplir sus actividades. A pesar de que el entorno ágil en el que se desarrollan las aplicaciones proporciona las herramientas para descubrir la mayor parte de los errores, muchos no se encuentran a tiempo y pueden causar problemas de mal funcionamiento, incidiendo en el cumplimiento de actividades de la empresa. Para solventar la situación descrita, se ha adoptado prácticas formales definidas en el marco de trabajo Capability Maturity Model Integration (CMMI), que permitan mejorar y escalar los procesos del área de pruebas. Este trabajo se ha desarrollado en una empresa real que provee software como servicio; por lo cual ha sido posible analizar los registros de fallos registrados durante un año, a fin de determinar las causas más comunes de las fallas experimentadas y descubrir su contraparte dentro de CMMI.

Abstract

This study presents an alternative that can be applied to improve the area of software testing within companies dedicated to providing software as a service. At present, the applications provided are critical for the proper functioning of companies, so if they present problems, they will cause companies not to be able to comply with their activities. Although the agile environment in which applications are developed provides the tools to discover most of the errors, many of them are not found in time and can cause malfunctioning problems, affecting the compliance of business activities. In order to solve this situation, formal practices defined in the Capability Maturity Model Integration (CMMI) framework have been adopted to improve and scale the processes in the testing area. This work has been developed in a real company that provides software as a service, so it has been possible to analyze the records of failures recorded during a year, in order to determine the most common causes of the failures experienced and discover its counterpart within CMMI.

Índice general

Índice general	xi
Índice de figuras	xiii
Índice de tablas	xv
1 Introducción	1
1.1 Antecedentes	2
1.1.1 Descripción de la empresa	2
1.1.2 Manejo de casos	4
1.2 Planteamiento del problema	4
1.3 Objetivos	5
1.3.1 Objetivo general	5
1.3.2 Objetivos específicos	6
1.4 Marco teórico	6
1.4.1 Descripción	6
1.4.2 Componentes	7
1.4.3 Enfoques de implementación	7
1.4.4 Justificación	7
1.5 Revisión sistemática de literatura	8
1.5.1 Resultados de la revisión sistemática de literatura	9
1.5.2 Análisis de datos	10
1.5.3 Contribución al trabajo actual	12
2 Metodología	13
2.1 Procedimientos utilizados	14
2.1.1 Evaluación de los casos	15
2.1.2 Selección del modelo para el análisis de costos	15
2.1.3 Identificación de patrones	15
2.1.4 Identificación de las áreas de proceso CMMI	16
2.2 Ejecución de Procedimientos	16

2.2.1	Análisis de defectos	16
2.2.2	Análisis de costos	18
2.2.3	Análisis de Patrones	20
2.2.4	Análisis de las áreas y procesos de la empresa	20
2.2.5	Análisis de métricas	20
2.2.6	Análisis de áreas de proceso CMMI	21
2.3	Propuesta de implementación	22
2.3.1	Áreas de proceso CMMI	22
2.3.2	Planificación de la implementación	24
2.3.3	Plan de implementación	25
2.3.4	Métricas	26
3	Análisis de resultados	27
3.1	Áreas o procesos fallidos	27
3.2	Costos asociados a los casos críticos	27
4	Plan de Implementación	31
5	Conclusiones y Recomendaciones	43
5.1	Conclusiones	43
5.2	Recomendaciones	43
A	Estructura de la base de datos de errores	45
	Bibliografía	49
	Siglas	55

Índice de figuras

1.1	Personas - Procedimientos - Herramientas	1
1.2	Flujo simple de desarrollo	5
2.1	Pasos metodología	14
2.2	Personal - Actividades	19
2.3	Muestra del plan implementación	26

Índice de tablas

1.1	Criticidad de incidentes según la empresa.	3
1.2	Comparación entre niveles de capacidad y madurez.	8
1.3	Artículos en bases de datos indexadas.	10
2.1	Estadísticas 2017	20
3.1	Procedimientos fallidos.	28
3.2	Casos críticos por procedimiento.	29
3.3	Procedimientos fallidos recurrentes.	29
3.4	Costos de los grupos de recursos.	30
3.5	Costos de las actividades.	30
A.1	Elementos de la base de datos DataCasosCriticos.	45
A.2	Datos que almacena la tabla PasoProduccion.	46
A.3	Datos que almacena la tabla CasoCritico.	47
A.4	Datos que almacena la tabla Clasificacion.	47
A.5	Datos que almacena la tabla ClasificacionCasoCritico.	48

Capítulo 1

Introducción

Todas las organizaciones abocadas a la tarea de desarrollar software, ya sea para terceros o para su propio beneficio, afrontan el desafío de la mejora continua para poder ser competitivos y poder entregar un servicio acorde con la realidad actual de la sociedad [1]. Esto implica el mejoramiento de los procesos, de las herramientas, y de los equipos de personas que producen dicho software (Figura 1.1). La estrategia utilizada para este mejoramiento puede ser muy variada [2, 3, 4, 5, 6, 7, 8, 9, 1], la misma que puede ir desde la identificación de puntos de mejora, el uso de metodologías formales, el uso de metodologías ágiles, o el uso o adaptación de uno o varios componentes de las ya mencionadas. Por lo que el método que se utilice dependerá de la situación actual de la empresa [10].

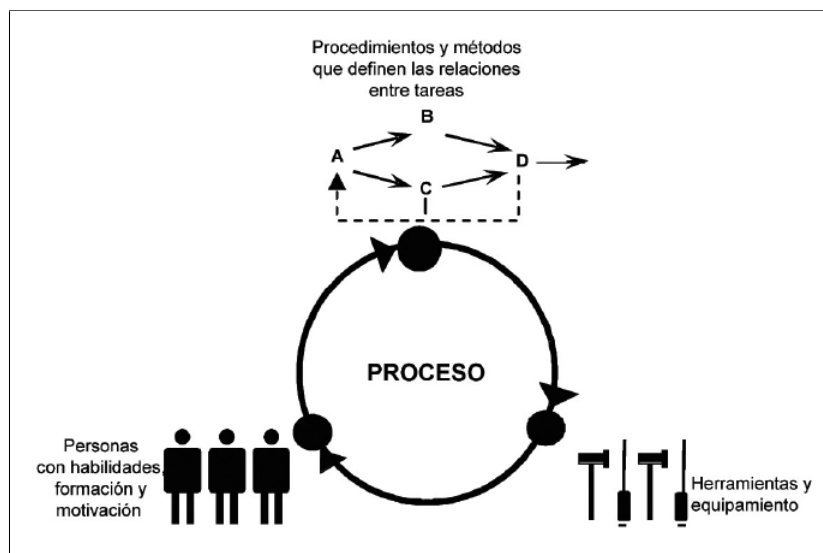


Figura 1.1: Software Engineering Institute (SEI) - Dimensiones de enfoque para mejorar la actividad de la empresa [1].

1.1 Antecedentes

1.1.1 Descripción de la empresa

El trabajo de investigación se realizó para una de las empresas líderes en servicios para clínicas y hospitales de diálisis en Estados Unidos. Su enfoque está en crear relaciones sólidas y duraderas con sus clientes, entendiendo sus necesidades y ayudándoles a superar los retos que enfrentan día a día en el ámbito de la diálisis.

Los servicios que ofrece son:

- Software para la administración de los procesos asociados al tratamiento de diálisis, que es un conjunto de aplicaciones web catalogadas como “Electronic Health Record (EHR)” [11].
- Servicios de análisis de datos.
- Servicios extendidos:
 - Facturación de insumos asociados a los procesos del tratamiento de diálisis.
 - Verificación de información para la integración con sistemas de terceros.
 - Cumplimiento y envío de datos a “Consolidated Renal Operations in Web-enabled Network (CROWNWeb)” [12].
- Servicios de administración:
 - Incluye los servicios extendidos.
 - Administración operativa de la clínica.
 - Administración de protocolos de medicinas.
 - Administración de inventarios.

Su producto estrella es un sistema EHR que se ofrece al mercado a través del modelo de Software como Servicio “Software as a Service (SaaS)” que se basa en la comercialización y uso de software enteramente a través de internet [13]. El modelo SaaS en sí propone algunas variantes, dentro de las cuales la empresa en cuestión adopta el alojamiento de varios clientes dentro de su infraestructura, además de utilizar el mismo software para todos los clientes en lugar de crear una instancia por cliente [13]. La empresa requiere establecer los parámetros que permitan mejorar la corrección de defectos y entregar nuevas funcionalidades y, al mismo tiempo, el cliente pueda ser protegido y tenga espacio para hacer valer sus derechos.

Estos parámetros se establecen a través de un acuerdo de nivel servicios “Service Level Agreement (SLA)” [14], mismo que la empresa firma con cada uno de sus clientes. El SLA que la empresa utiliza, cuenta con los siguientes términos:

- La disponibilidad de personal en caso de asistencia técnica.
- El tiempo de disponibilidad total de la aplicación en el año.
- Las compensaciones en caso de la indisponibilidad de las aplicaciones.
- Las consideraciones sobre defectos.
- Las clasificaciones de defectos según su criticidad, que de acuerdo a la empresa, es la dificultad que tenga un cliente para utilizar el software o su necesidad de requerimientos adicionales como se muestra en la Tabla 1.1.
- Y, finalmente los tiempos de respuesta según criticidad.

Criticidad	Definición
Crítico	Un problema reportado evita que más de un empleado del cliente pueda ejecutar un procedimiento requerido inmediatamente.
Alto	Un problema reportado evita que un empleado del cliente pueda ejecutar un procedimiento requerido, con la excepción de que el resto del personal si pueda utilizar otros componentes del sistema.
Medio	- Defectos del sistema NO-Críticos. - Configuraciones que el cliente necesita modificar.
Planificado	- Modificaciones NO-Urgentes de funcionalidades existentes. - Reparación de descuidos. - Consulta de datos del sistema que no se encuentran en la interfaz de usuario. - Solicitudes extendidas para entrenamientos.
Proyectos	Petición reportada de nueva funcionalidad o mejoramiento de la funcionalidad existente.

Tabla 1.1: Criticidad de incidentes según la empresa.

Creando así un marco de trabajo, en el que, la empresa está comprometida en ganar la confianza de sus clientes, compartiendo el riesgo en la adopción de su tecnología y ofreciendo productos de alta calidad.

1.1.2 Manejo de casos

Todas las comunicaciones de los clientes con la empresa son registradas en un sistema de gestión de relaciones de clientes “Customer Relationship Management (CRM)” [15], para ser documentadas, catalogadas, y priorizadas según la Tabla 1.1. Algunos de los casos como configuraciones, soporte y, requerimientos de información, son resueltos por el equipo de atención al cliente.

Por otro lado, los requerimientos para cambios, correcciones, o proyectos son entregados al equipo de análisis de negocios “Business Analysis (BA)”, que re-prioriza y agenda su ejecución conjuntamente con el equipo de “Desarrollo”. El seguimiento de estos casos es llevado en un sistema para la administración del ciclo de vida del desarrollo de aplicaciones “Software Development Life Cycle (SDLC)” [16].

El proceso se inicia con la especificación de un requerimiento por parte del equipo de BA, el cual que es entregado al equipo de desarrollo, que internamente se compone de equipos de programación y pruebas, con un equipo adicional de soporte para configuraciones e instalación de aplicaciones. El equipo de programación entrega incrementalmente el desarrollo de los requerimientos al equipo de pruebas. Si el equipo de pruebas encuentra defectos en las aplicaciones los reporta para que se codifiquen los cambios, por lo que se crea un ciclo entre los equipos de programación y pruebas que se repite hasta que se satisfagan los requerimientos. Finalmente el requerimiento es devuelto al equipo de BA para que este certifique el cumplimiento de lo inicialmente especificado. Todo esto dentro del marco de metodologías ágiles como Kanban [17] o Scrum [18]. El flujo simple de un requerimiento se muestra en la Figura 1.2

1.2 Planteamiento del problema

Debido a que el principal modelo utilizado por parte de la empresa es SaaS, los cambios (nuevos requerimientos o mantenimiento) que se vayan incorporando al entorno de producción impactarán a todos los clientes. Esto abre la posibilidad de que al introducir un defecto en las aplicaciones, uno o varios clientes se vean afectados por el mismo. De entre los criterios mostrados en la Tabla 1.1, los casos críticos son los que más riesgo conllevan.

Cada caso crítico evita que empleados de uno o varios clientes dejen de utilizar el sistema; además, si dicho incidente no se puede resolver en el tiempo establecido por el SLA, se tiene como consecuencia concesiones en el servicio a favor del cliente, o incluso la terminación del contrato entre la empresa y el cliente. Debido a esto el análisis de este tipo casos cobra importancia para la empresa.

El presente trabajo busca mejorar los procesos del área de pruebas con prácticas definidas en

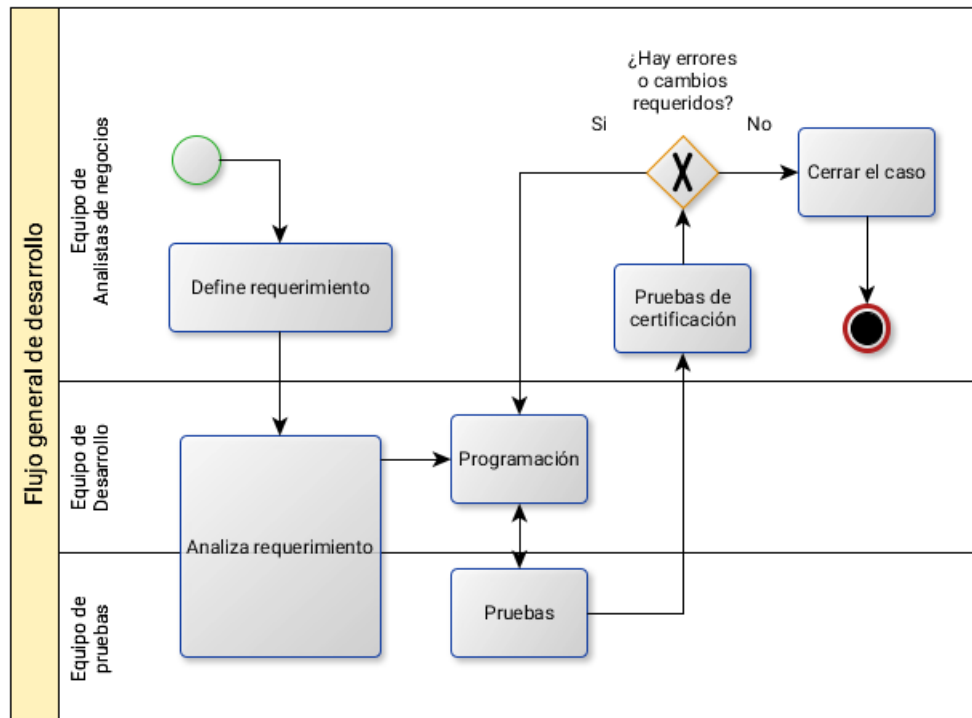


Figura 1.2: Flujo de desarrollo en una iteración.

CMMI, para poder encontrar más defectos durante las fases iniciales de desarrollo, y minimizar la incidencia de casos críticos reportados en producción.

1.3 Objetivos

1.3.1 Objetivo general

Proponer un plan de adopción y adaptación de al menos tres áreas de proceso CMMi dentro de equipos ágiles de pruebas.

El presente estudio se enfoca en el análisis de los fallos críticos reportados por clientes, con la finalidad de encontrar patrones que permitan identificar las áreas de mejora dentro de la empresa. Una vez identificadas estas áreas, se emparejan las mismas con prácticas pertenecientes a Capability Maturity Model Integration (CMMI). Como resultado se propondrá un agenda para la adopción de dichas prácticas, adaptándolas a la realidad de la empresa.

Cabe aclarar, que no se busca la implementación de CMMI en la empresa. Lo que se desea es encontrar las prácticas dentro de CMMI que más le convengan implementar a la empresa en su situación actual, para poder obtener el beneficio de dichas prácticas.

1.3.2 Objetivos específicos

1. Realizar un análisis causa-efecto sobre 50 % de las fallas críticas registradas durante el año 2017 dentro de la empresa.
2. Realizar un análisis de costos sobre 50 % las fallas críticas registradas durante el año 2017 dentro de la empresa.
3. Encontrar patrones basados en el análisis de las fallas críticas.
4. Seleccionar los “procesos candidatos para mejoramiento” que se hayan identificado como inefectivos e ineficientes basados en los patrones encontrados.
5. Identificar las métricas necesarias para futuras evaluaciones de procesos.
6. Identificar las áreas de proceso en CMMi que tengan relación con los “procesos candidatos para mejoramiento”.
7. Identificar las prácticas adicionales para adoptar y adaptar las áreas de proceso CMMi en los procesos candidatos para mejoramiento”
8. Proponer un plan de implementación de las prácticas que permitan cumplir las áreas CMMi identificadas.

1.4 Marco teórico

La siguiente introducción tiene como fin de proporcionar un entendimiento básico de CMMI para su futura referencia en el resultado de la investigación.

1.4.1 Descripción

Capability Maturity Model Integration (CMMI) es un marco de trabajo que se enfoca en los procedimientos de una empresa, y en los resultados de su ejecución. Esto, proporciona la suficiente flexibilidad a las empresas que implementen CMMI para que puedan utilizar métodos adaptados a sus necesidades que les permitan cumplir las metas propuestas por el marco de trabajo. Al momento existen tres “sub divisiones” de CMMI, que son, CMMI para desarrollo, CMMI para adquisiciones, y CMMI para servicios. De los cuales, CMMI para desarrollo (CMMI-DEV V1.3) es la utilizada como referencia en esta investigación.

1.4.2 Componentes

Los elementos de CMMI están clasificados de la siguiente manera:

1. Nivel (grado de madurez) de la organización:
Una organización puede estar clasificada dentro de uno de los 5 niveles que CMMI define. Estos son (1) Inicial, (2) Gestionado, (3) Definido, (4) Gestionado cuantitativamente, y (5) Optimizado. Cada nivel tiene asociadas un grupo de áreas de proceso.
2. Áreas de proceso:
Un área de proceso es un grupo de prácticas relacionadas entre sí, que al ser implementadas en conjunto, deben satisfacer las metas del área de proceso.
3. Metas:
Es el resultado final que se debe cumplir según cada área de proceso. Para lo cual se deben seguir las diferentes prácticas definidas.
4. Prácticas:
Son actividades que se deben seguir para poder llegar a la meta planteada. CMMI describe lo que se debe ejecutar, mas no el cómo.
5. Sub-Prácticas:
Son descripciones detalladas para poder interpretar o implementar las prácticas descritas.

1.4.3 Enfoques de implementación

Existen dos enfoques que una organización puede seguir para implementar las áreas de proceso. El primero es un mejoramiento incremental de los procesos de la o las áreas que la organización haya seleccionado (“continuo”), y el segundo es optar por el mejoramiento de un conjunto de procesos relacionados que se distribuyan a través de la organización (“por etapas”). El enfoque “continuo” muestra el nivel de capacidad de los procesos, mientras que el enfoque “por etapas” muestra el nivel de madurez de la empresa. En cualquiera de los dos casos, la organización debe satisfacer todas las metas de las áreas de proceso involucradas del enfoque elegido [1]. En la Tabla 1.2 se muestra una comparación de los niveles de capacidad y madurez en referencia a los niveles de cada caso.

1.4.4 Justificación

CMMI se puede utilizar en cualquier campo del quehacer humano, por lo que empresas de diferentes tamaños pueden adoptarla según sus necesidades. Es decir, se puede evaluar a la empresa para saber su nivel actual y, según sus necesidades, planificar su evolución al nivel deseado o incluso resolver si este camino le conviene o no a la empresa [19]. Los motivos por los que una empresa elija CMMI como base para su mejoramiento son variados y dependen su situación.

Nivel	Enfoque “continuo” Niveles de capacidad	Enfoque “por etapas” Niveles de madurez
Nivel 0	Incompleto	
Nivel 1	Realizado	Inicial
Nivel 2	Gestionado	Gestionado
Nivel 3	Definido	Definido
Nivel 4		Gestionado cuantitativamente
Nivel 5		En optimizacion

Tabla 1.2: Comparación entre niveles de capacidad y madurez.

CMMI le permitirá a una empresa mejorar el desempeño de sus procesos y procedimientos organizacionales, mejorar su capacidad de entrega de productos o servicios de manera consistente y predecible, saber su situación en comparación de otras empresas que la implementen.

Cabe recalcar que el implementar CMMI requiere:

- Compromiso de los directivos de la empresa.
- Inversión monetaria por el costo de las evaluaciones. Esto depende de si es que se quiere obtener la certificación formal para la empresa, u obtener únicamente los beneficios de la implementación.
- Inversión en la capacitación del personal involucrado en la implementación de los cambios.
- Inversión en el tiempo y personal requeridos para la implementación de los cambios.

1.5 Revisión sistemática de literatura

Para la búsqueda de bibliografía pertinente, se ha utilizado un enfoque “centrado en el concepto” [20]. La finalidad de utilizar este enfoque es encontrar publicaciones relacionadas con el tema del presente trabajo, de tal manera que estas permitan esclarecer el aporte del mismo.

La revisión sistemática de literatura busca responder a las siguientes preguntas de investigación:

1. ¿Se pueden analizar los fallos críticos de tal manera que se puedan identificar patrones en las causas de los mismos?
2. ¿Se pueden analizar fallos críticos de tal manera que se puedan identificar prácticas o procesos, que no se estén tomando en cuenta en el desarrollo de software actual?

El proceso de revisión consistió de los siguientes pasos: Selección de las bases de datos indexadas en las que se realiza la búsqueda, determinación de las cadenas de búsqueda y determinación de los tópicos de relevancia.

Bases indexadas: Se realizó la búsqueda en las siguientes bases de datos indexadas: Springer, IEEE Xplore, ACM, y ScienceDirect.

Cadenas de búsqueda: Las cadenas de búsqueda utilizadas tiene como objetivo evidenciar las áreas de prevención de defectos y mejoramiento del desarrollo de software. Por lo que, se utilizaron las siguientes combinaciones en las diferentes bases de datos indexadas:

- “defect prevention” AND “software process improvement”
- “defect causal analysis” AND “software”
- “defect analysis” AND “software process improvement”

Los diferentes criterios para filtrar los trabajos de investigación fueron:

1. El año de publicación se filtró de tal manera que el año de publicación del trabajo de investigación debía ser igual o mayor al año 2012. Se consideró un horizonte de 5 años debido a la naturaleza cambiante de las tecnologías de la información.
2. La cadena de investigación debe estar presente en las propiedades: “Title”, “Abstract” y, “Keywords” de los trabajos de investigación.
3. Se tomaron en cuenta los trabajos de investigación de tipo “Artículo científico” publicados en proceedings de conferencias o revistas relacionados al tema de investigación. No se tomaron en cuenta los trabajos de investigación que sean de tipo “Libro” o “Capítulos de libro”.

Selección de tópicos: Los trabajos de investigación seleccionados han aportado a la validación de la investigación actual. Los tópicos que estos trabajos discuten son:

- Mejoramiento de los procesos de desarrollo de software;
- Agilidad en el proceso de desarrollo de software;
- Madurez de los procesos de desarrollo de software; y,
- Análisis y clasificación de defectos.

1.5.1 Resultados de la revisión sistemática de literatura

Los resultados de la revisión de literatura se presentan a través de un resumen de los trabajos de investigación seleccionados a partir de las búsqueda en bases de datos indexadas, una síntesis de los diferentes trabajos de investigación, y su relación o aporte con la investigación actual.

Los resultados de la búsqueda de trabajos de investigación se muestra en la Tabla 1.3.

“defect prevention” AND “software process improvement”							
Fuente	2012	2013	2014	2015	2016	2017	2018
Springer	2				1		
IEEE Xplore	1						
ACM		1		2	1	1	
ScienceDirect	1		1	2	1		
Total	4	1	1	4	3	1	
“defect causal analysis” AND “software”							
Fuente	2012	2013	2014	2015	2016	2017	2018
Springer			2				
IEEE Xplore	2						
ACM						1	
ScienceDirect		1					
Total	2	1	2			1	
“defect analysis” AND “software process improvement”							
Fuente	2012	2013	2014	2015	2016	2017	2018
Springer							
IEEE Xplore		1		1	1		1
ACM						1	
ScienceDirect		1					
Total		2		1	1	1	

Tabla 1.3: Artículos en bases de datos indexadas.

1.5.2 Análisis de datos

Síntesis de hallazgos

En resumen, los artículos encontrados durante la búsqueda se relacionan con las ideas del trabajo actual. Confirmando así la necesidad del análisis de los defectos como una de las técnicas de mejoramiento continuo [21, 22, 8], así como el análisis de datos históricos de defectos para mejorar los procesos de pruebas [23, 22, 7]. Los tópicos a los cuales los diferentes artículos se refieren son:

- Identificar problemas, patrones y oportunidades de mejora.
Los trabajos coinciden sobre el análisis que se deben hacer sobre los eventos que han causado un impacto importante. Centrando sus esfuerzos en muestras estadísticas con la finalidad de encontrar oportunidades de mejora.

Se denota adicionalmente que, independientemente de las técnicas utilizadas, la gran parte de las empresas no llevan a cabo un adecuado proceso de mejoramiento basado en

el aprendizaje de los errores cometidos [21]. Sin embargo la aplicación efectiva de DCA (Defect causal analysis) [24, 25] ha ayudado a reducir la tasa de defectos encontrados en un 50 % [26, 27]. Además de DCA, existen otros frameworks/técnicas como SPI (ver ejemplo en [4]), Software Process Reflexivity (ver ejemplo en [5]), DevOps [6] (ver ejemplo en [3]) que, son otros caminos que se pueden seguir o adoptar para llegar al objetivo mayor que es el de mejorar el desarrollo de software.

- **Costos de los procesos de pruebas.**
Los procesos de pruebas constituyen más del 50 % del costo total del desarrollo de software [28], esto, junto con el hecho de que la complejidad del software se va incrementando con el tiempo, hace que los costos de las pruebas se incrementen de igual manera [28]. Dando paso a la necesidad de encontrar procesos o modelos de pruebas más efectivos [28], de tal forma que cualquiera que sea el proceso que se utilice, este, sea sometido a procesos de mejoramiento continuo con la finalidad de disminuir los costos asociados [10].
- **Adopción de modelos o frameworks de prueba.**
Los equipos ágiles tienen dificultad en asimilar o implementar modelos o frameworks formales, ya que intrínsecamente, éstos, no se guían por procesos prescriptivos [9, 29].
- **Análisis y priorización de defectos.**
Se puede extraer valiosa información acerca del aseguramiento de la calidad que se puede brindar a un sistema mediante el análisis de los incidentes que se han levantado por parte de los usuarios del sistema reportando los defectos que éste tiene [30]. Adicionalmente, la calidad de cualquier sistema también está determinada por la calidad del servicio proporcionado sobre los incidentes inesperados que ocurren en el. Esto sin embargo, no puede suceder sin que se ejecute un proceso sistemático de recolección de los datos sobre los defectos, lo cual debe ser implementado por todo tipo de organizaciones, incluso aquellas poco maduras [24].
- **Predicción de bugs.**
Un área que el presente trabajo no toma en cuenta es la predicción de defectos. La misma se encuentra expandida en varios de los trabajos encontrados [31, 32, 33, 30]. Las técnicas empleadas permiten identificar módulos potencialmente defectuosos sobre los cuales se debería llevar a cabo una revisión más minuciosa, especialmente en etapas tempranas del ciclo de vida de las aplicaciones [31].
Con lo que estudios previos pueden ayudar a identificar posibles incidentes, que de no ser debidamente gestionados corren el riesgo de desembocar en accidentes [32].
Otro de los tópicos que se exploran para la predicción de defectos es el uso de métricas, que con el debido análisis, permite identificar la relación que existe entre estas y la ocurrencia de bugs [33].
- **Clasificación y reporte de defectos.**
Los trabajos encontrados tienen un componente común relacionado con la clasificación

de defectos. Esta práctica cobra gran importancia al poder reducir los costos y el tiempo empleados en el análisis de los mismos [34, 8]. Una de las formas de comunicar esta información es a través de reportes y, el detalle de estos reportes de defectos extienden el alcance y la calidad de los análisis estadísticos, además proveen una ampliación significativa en la evaluación y refinamiento de los procesos de desarrollo de software y, una predicción confiable de defectos [35].

1.5.3 Contribución al trabajo actual

Los trabajos encontrados remarcan los beneficios de analizar los defectos con el fin de mejorar los productos y, los procesos con los cuales estos productos fueron desarrollados. En todos los trabajos encontrados se realizan preguntas para encontrar datos, pistas, o mejores procesos que ayuden a localizar la fuente de los problemas. El trabajo actual tiene el mismo objetivo, de tal manera que analizar los defectos críticos que la empresa ha enfrentado permita encontrar procesos dentro de CMMI [1] que se ajusten a los vacíos que estos defectos han evidenciado, y proponer un plan de adaptación de estos procesos/prácticas a la realidad de la empresa. Esta es una de las formas en las que empresas de diferentes tamaños han podido mejorar la calidad del software producido, además de acelerar el desarrollo de software. [4]

Capítulo 2

Metodología

El presente trabajo se basa en la metodología de casos de estudio, la cual se enfoca en un caso específico al sujeto de la investigación, y por lo tanto se está limitado al caso en cuestión. Las aplicaciones de esta metodología son diversas y puede ser utilizada en estudios de problemas organizacionales y empresariales, educativos, estudios familiares, conflictos internacionales, desarrollo de tecnología, e investigación de problemas sociales, y más comúnmente, en investigaciones de políticas públicas, negocios y administración pública. Dentro de un contexto de evaluación, la metodología es utilizada para documentar y analizar procesos de implementación. Por lo que, esta ha sido asociada con la evaluación de procesos.

La aplicación de esta metodología conlleva la ejecución de actividades preliminares como la búsqueda de la literatura apropiada, la definición de una unidad de análisis, identificar el criterio para seleccionar y detectar casos potenciales, y sugerir variables de interés. Tomando en cuenta que el método favorece la emulación del método científico, este incluye los siguientes pasos [36]:

- Plantear preguntas explícitas de investigación.
- Desarrollo de un diseño de investigación formal.
- Usar la teoría e investigaciones previas para desarrollar hipótesis a favor y en contra la investigación.
- Recolectar datos empíricos para probar estas hipótesis.
- Crear una base de datos que registre los datos sin preferencia o sesgo a favor o en contra de la investigación.
- Realizar un análisis cuantitativo, cualitativo, o ambos dependiendo del tópico e investigación del diseño.

Dentro de este contexto, el trabajo actual utiliza estos conceptos para el estudio de los defectos críticos que la empresa ha enfrentado y resuelto en favor de sus clientes, con la intención de encontrar las causas subyacentes, aprender de estos errores, y poner en práctica acciones que minimicen la ocurrencia de este tipo de defectos. Los pasos que se tomaron en base a la metodología se pueden revisar en la figura 2.1.

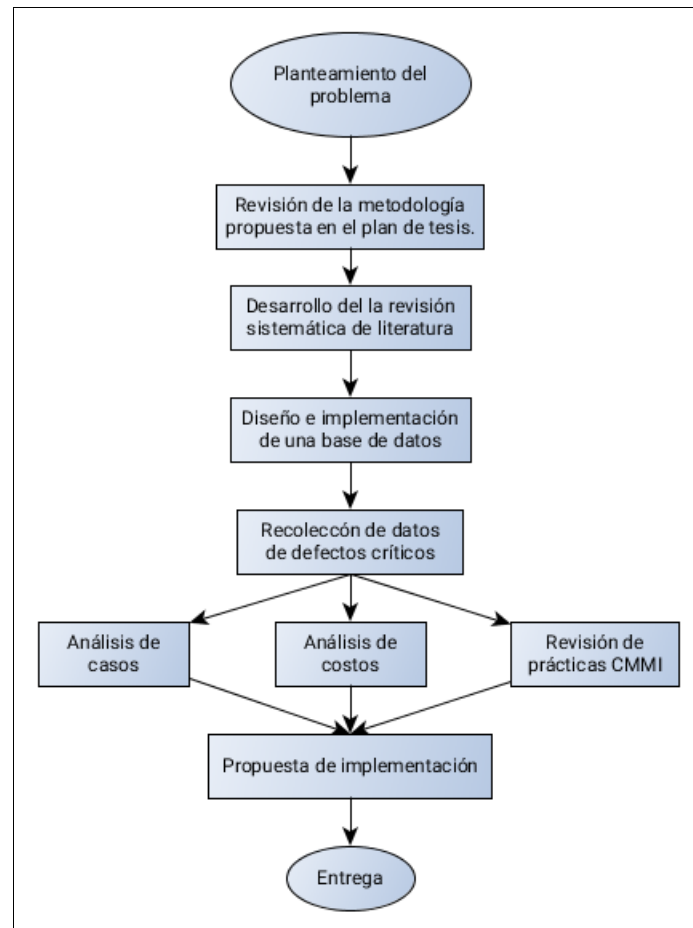


Figura 2.1: Pasos de la metodología.

2.1 Procedimientos utilizados

Los procedimientos utilizados durante la ejecución de la investigación son los siguientes:

2.1.1 Evaluación de los casos

Para la evaluación de los diferentes casos se utilizó el método de “Análisis Causa-Raíz” esta, es una metodología sistemática y reactiva, que ayuda a identificar las causas de un problema y por tanto, a evitar que sea recurrente. Los pasos realizados para este análisis [37] son:

1. Establecer el problema que será analizado del caso crítico.
2. Lluvia de ideas para recolectar causas específicas del caso crítico.
3. Revisar y discutir las causas específicas del caso crítico.

2.1.2 Selección del modelo para el análisis de costos

Se ha elegido utilizar el modelo basado en actividades “Sistema de costes ABC (Activity-Based Costing)” [38] para el análisis de defectos críticos, debido a que los clientes tienden a pagar más por requerimientos específicos que por un paquete completo. Eso hace que se necesiten estimaciones más precisas y un mejor seguimiento de las actividades, con lo que se puede enfocar en el tamaño y esfuerzo del proyecto como una unidad. La diferencia está en asignar costos a actividades o procesos, mas no a departamentos [10]. El análisis de costos basado en actividades es un modelo que permite encontrar el costo de un producto o servicio asociándolo con las actividades que se necesitaron para completarlo, y ya que, una actividad utiliza varios recursos de una empresa, el costo de la actividad se encuentra tras calcular el porcentaje de uso de estos recursos por parte de la misma.

El modelo ABC [38], se ajusta a la empresa, por lo que se deben ejecutar las siguientes faces: análisis de costos, y estimación usando el modelo ABC.

1. Análisis de costos: Se necesita que se identifiquen los costos de los recursos a través de la revisión de la contabilidad, y la duración de las actividades.
2. Aplicación del modelo ABC: Se realizan los cálculos con los parámetros obtenidos.

2.1.3 Identificación de patrones

La identificación de patrones parte de analizar la información de los diferentes casos basados en sus características y, poder recopilar tal información de diferentes fuentes dentro de la empresa tales como CRM, SDLC, correos electrónicos, y conversaciones con el personal a cargo.

Por lo que, en lugar de utilizar documentos para registrar esta información (procesadores de palabras, hojas de cálculo), se optó en crear una base de datos de errores similar a una “Known Error Database (KEDB)” [39], que es una práctica de “Information Technology Infrastructure Library (ITIL)” [39]. Las diferencias principales con la KEDB de ITIL, es que está no registra todos los incidentes, sino únicamente la información pertinente a defectos críticos, basados en

los criterios de la Tabla 1.1. Además, podríamos decir que extraemos y registramos la metadata [40] de los casos, es decir, obtenemos los datos que describen datos.

Los objetivos de esta base de datos son los siguientes:

- Registrar la información de los casos.
- Recopilar información de varias fuentes en un solo repositorio.
- Clasificar los casos en base a la evaluación de los mismos.
- Agregar propiedades a los casos según se vayan encontrando características que lo requieran en base al análisis de las fuentes.
- Permitir un acceso ágil a los datos de los casos en base a las características generadas.

2.1.4 Identificación de las áreas de proceso CMMI

El motivo por el cual se eligió CMMI como base para el mejoramiento de las áreas dentro de la empresa, es debido a su efectividad en desarrollo de software y a su amplia utilización en el mercado. Adicionalmente, el enfoque de procesos de CMMI está alineado con la forma de trabajar dentro de la empresa, esto hace que el mejoramiento y adopción de procesos sea natural dentro de la organización. Finalmente, ya que el estudio se enfoca en procesos específicos para el área de pruebas de la empresa, se ha optado por usar el enfoque de capacidad en lugar del de madurez sobre las áreas de proceso, como se explica en la sección 1.4.3.

La selección de áreas de proceso CMMI deben estar acorde con los procesos de pruebas susceptibles de mejora. Para esto, inicialmente se deben identificar dichos procesos, evaluando el entorno en los que éstos se ejecutan; y, posteriormente, seleccionar las de áreas de proceso que proporcionen las prácticas adecuadas para los procesos de la organización.

2.2 Ejecución de Procedimientos

Lo procedimientos mostrados a continuación son el resultado de la recopilación de información y el apoyo del personal dentro de la empresa. El producto de la empresa sobre el cual se realiza el análisis es su software EHR, el mismo se ofrece al mercado mediante el modelo SaaS.

2.2.1 Análisis de defectos

Se analizaron los casos críticos basados en el método causa-raíz. Este proceso se inició con la recopilación de información sobre una muestra de 31 casos críticos de los 83 ocurridos en el año 2017. La muestra se recopiló mediante el uso de muestreo no probabilístico debido a que la diversidad de los casos no representaban consistentemente a toda la población de casos críticos.

La información se obtuvo de los sistemas internos CRM, SDLC, y correos electrónicos.

Desarrollo de una solución para el caso crítico

La implementación de un caso crítico sigue el procedimiento regular de desarrollo, con la característica de que es prioritario para los equipos responsables. La corrección de un caso crítico implica, que un grupo de personas de varios equipos paren sus actividades para dedicarse exclusivamente al desarrollo de una solución. Al iniciar la codificación del caso se inicia paralelamente la planificación para su paso a producción, una vez que el caso a sido puesto en producción se culmina el proceso con la comunicación al cliente.

Fuentes de información para el análisis de los casos críticos

La información se extrajo de los siguientes procedimientos en los que, cada grupo responsable del mismo cuenta con la información necesaria sobre los pasos ejecutados en su gestión del caso:

- Manejo del caso por parte del equipo de soporte (CRM).
- Resolución del caso, desde la definición de requerimientos, hasta su aprobación (SDLC).
- Plan del paso a producción y ejecución del mismo.
- Comunicaciones entre los diferentes equipos (correos electrónicos).

Proceso de recopilación de información

Los pasos que se siguieron para el análisis de defecto fueron:

1. Listar todos los casos críticos del año 2017.
2. Tomar una muestra de entre estos casos críticos.
3. Por cada caso crítico:
 - (a) Buscar la información relacionada en el sistema CRM.
 - (b) Obtener el tiempo de resolución del caso CRM.
 - (c) Buscar la información relacionada en el sistema SDLC.
 - (d) Analizar el requerimiento del caso.
 - (e) Analizar el criterio de aceptación del caso.
 - (f) Identificar el defecto asociado.
 - (g) Identificar la o las causas que provocaron el caso.
 - (h) Identificar la o las acciones que se tomaron para su resolución.
 - (i) Evaluar el código fuente (que resolvió el error) para identificar si se produjo después a un paso a producción.

-
- (j) Buscar la información relacionada en el correo electrónico que pueda ser de aporte para la resolución del caso.
 - (k) Evaluar toda la información recopilada para identificar el proceso o procesos que fallaron.

La base de datos se fue alimentando durante el proceso de evaluación de cada caso, esto significa que, por cada caso se creó un registro en la tabla de base de datos “CasoCritico” (detalle de datos en la Tabla A.3). Adicionalmente, esta evaluación permitió identificar el o los procesos de la empresa que fallaron en encontrar los errores que desembocaron en el defecto en producción, por cada proceso se creó un registro en la tabla de base de datos “Clasificacion” (detalle de datos en la Tabla A.4). Finalmente, se necesitó relacionar el caso crítico con el proceso fallido dentro de la empresa, esta relación se registró en la tabla de base de datos “ClasificacionCasoCritico” (detalle de datos en la Tabla A.5), se requirió esta última debido a que uno o varios procesos pueden ser responsables de los errores y, al mismo tiempo un proceso puede ser responsable de uno o varios casos críticos.

2.2.2 Análisis de costos

Los pasos ejecutados para el análisis de costos fueron:

1. Análisis de costos inicial: Los valores son tomados de los registros contables y de los sistemas internos CRM, y SDLC.
 - Análisis de los grupos de recursos y, costo total por recurso: Se identificaron los roles dentro de la empresa que están involucrados en la resolución de los casos críticos. Esta lista sirve como base para evidenciar los costos asociados con el personal, como se muestra en la Tabla 3.4.
 - Identificación de actividades: Se revisó el proceso actual para obtener una lista de tareas y agruparlas en actividades. Se analizaron las actividades para registrar el uso que estas hacen del personal, y así obtener una relación entre los costos globales de la resolución de casos críticos y el personal, mostrado en la Tabla 3.5.
 - Costo total por actividad y, costo de la actividad por día: Se calcularon los costos de las actividades ejecutadas para la resolución de casos críticos, para así obtener valores anuales y día-hombre por cada una de las mismas. Estos datos se pueden revisar en la Tabla 3.5.
2. Aplicación del modelo ABC: Se analizaron los casos para obtener los costos aproximados de los mismos.

La Figura 2.2 muestra las actividades que los diferentes roles ejecutan.

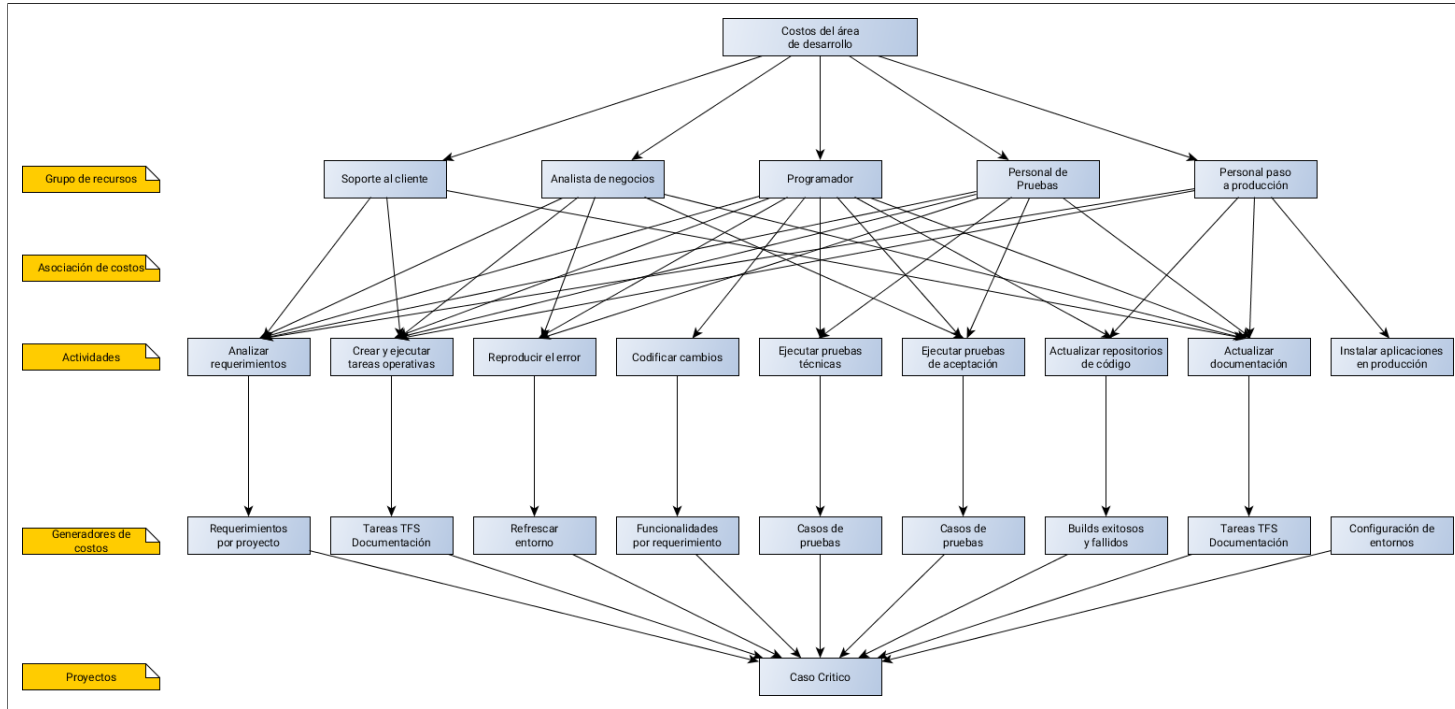


Figura 2.2: Personal asociado a las actividades-costos.

2.2.3 Análisis de Patrones

Para la recopilación de información y el análisis de los diferentes casos se requirió una herramienta que permita analizar la información recopilada usando su metadata [40]. Como se explicó en la sección 2.1.3, la herramienta necesita cumplir con varios requerimientos, por lo que se eligió la creación de una base de datos en “Microsoft Sql Server Express 2017”, y se la denominó “DataCasosCriticos”. El propósito de los diferentes objetos de base de datos se muestra en la Tabla A.1.

Los datos registrados se recopilaron dentro del siguiente contexto: *Después de que un caso haya pasado las diferentes fases de desarrollo y de certificación y, completada su planificación para “deploy”, es decir los pasos necesarios para la disponibilidad de su uso en producción [41], el equipo de BA evalúa el caso y, puede solicitar que se ponga en producción inmediatamente o que se espere por la completitud de otros casos críticos en proceso de resolución que cuenten con no nomas de 2 o 3 días para su finalización, para así ejecutar un solo deploy.*

2.2.4 Análisis de las áreas y procesos de la empresa

Para seleccionar los procesos de la empresa en los que se enfocará la propuesta, se analizaron los resultados de la base de datos en la cual se buscaron patrones en los incidentes. Este análisis mostró cuales son los procesos relacionados, o que afecten al equipo de pruebas, que pueden ser mejorados. Los mismos son: pruebas de regresión, casos de pruebas, y revisión de requerimientos. La lista completa de los procesos que necesitan ser mejorados dentro de la empresa está mostrada en la Tabla 3.1

2.2.5 Análisis de métricas

No existen métricas oficiales dentro de la organización, no obstante, en la Tabla 2.1 se muestran estadísticas¹ que pueden dar un punto de vista de alto nivel de la situación de la misma. Esto puede servir como una línea base de referencia para futuros análisis dentro de la organización.

Estadística	Valor
Número de pasos a producción en el año	84
Número de casos críticos en el año	70
Número de casos críticos promedio después de un paso a producción	$\frac{70}{84 - 70} = 5$

Tabla 2.1: Estadísticas 2017

¹La empresa lleva registros de cada evento de paso a producción en documentos desde el año 2014.

Sin embargo, se implementarán las métricas que se escojan basadas en los objetivos de la empresa, ya que las métricas no son un fin en si mismo, si no un medio de monitorear y conseguir dichos objetivos. Por lo que, cada métrica deberá contar con el objetivo explicito directamente ligado a los objetivos de negocio y a los atributos que seran medidos [42].

2.2.6 Análisis de áreas de proceso CMMI

En CMMI, tanto el enfoque “continuo” como en el enfoque “por etapas” son una adopción secuencial de prácticas o áreas de procesos respectivamente. Debido a esto, por ejemplo, la adopción de un área de proceso de nivel 5 sin previamente haber completado sus dependencias sería extremadamente complicado; por lo que, para maximizar los beneficios del marco de trabajo, sería necesario implementar estas dependencias en primer lugar [1]. Con tal antecedente, para esta implementación se han elegido áreas de proceso de nivel 2, con un grupo de dependencias bajo, y que cuyas práctica se alineen con los procesos de pruebas de la empresa.

A continuación se muestran las áreas de proceso seleccionadas, junto con los procesos de la empresa relacionados, y la manera en la que se ambas relacionan.

Planificación del proyecto (PP)

“El propósito de la Planificación de Proyecto (PP) es establecer y mantener planes que defina las actividades del proyecto.” [1]. La implementación de esta área de proceso permitirá, definir actividades destinadas al equipo de pruebas, para poder planificar, ejecutar, y dar seguimiento a las mismas de tal manera, que estas se alineen con los proyecto a ser desarrollados. Estas actividades se relacionan con las causas: “Casos de pruebas” y “Pruebas de regresión”, los mismos se encuentra en la Tabla 3.1.

Gestión de requisitos (REQM)

“El propósito de la Gestión de Requisitos (REQM) es gestionar los requisitos de los productos y los componentes de producto del proyecto, y asegurar la alineación entre esos requisitos, y los planes y los productos de trabajo del proyecto” [1]. Implementar prácticas de esta área de proceso permitirá gestionar los requerimientos y desarrollar el producto de acuerdo con sus requisitos, ya que, algunos de los requerimientos que los equipos recibieron para el desarrollo se encontraban incompletos o confusos, y esto desembocaron en casos críticos. La causa de casos críticos “Requerimientos” se encuentra en la Tabla 3.1.

Gestión de la configuración (CM)

“El propósito de la Gestión de Configuración (CM) es establecer y mantener la integridad de los productos de trabajo utilizando la identificación de la configuración, el control de la configuración, el informe del estado de la configuración, y las auditorías de la configuración.” [1]. Aunque la fuente los casos críticos tienen un bajo indice al tener como motivo un entorno

configurado incorrectamente o copias de código no sincronizadas (ver Tabla 3.2), es suficiente para implementar las prácticas necesarias dentro de esta área de proceso. Ya que, los entornos de pruebas dentro de la empresa son la herramienta básica e indispensable para que el equipo de pruebas pueda validar las aplicaciones que se pondrán en manos del cliente. De la misma, al ser el código fuente uno de los activos más importantes para la empresa, la correcta administración de los cambios que reciba tiene un impacto directo en la calidad de las aplicaciones que se construyen y distribuyen en base a este. Las causas de casos críticos “Administración de cambios de código”, y “Entornos” se encuentran en la Tabla 3.1.

2.3 Propuesta de implementación

La propuesta de implementación se centra en las áreas de proceso seleccionadas para mejorar los procesos asociados a los equipos de pruebas dentro de la empresa.

2.3.1 Áreas de proceso CMMI

La sección actual, muestra los detalles (tomados de [1]) de las áreas de proceso seleccionadas en el estudio. Éstas son las metas y prácticas que se desean implementar, adaptar o descartar según la situación actual de la empresa. Por ejemplo, la práctica específica “1.4 Estimar el esfuerzo y el coste” del área de proceso “Planificación del proyecto (PP)” no se puede implementar en su totalidad, ya que el área de pruebas no maneja ningún tipo de proceso de costos. Por otro lado la implementación de la práctica específica “1.1 Estimar el alcance del proyecto” en la misma área de proceso, proporcionaría una visibilidad de alto nivel del proyecto que se quiere desarrollar, esta es una herramienta con la que el equipo de pruebas no cuenta en la actualidad.

Este tipo de evaluación se realizará durante la ejecución del plan de implementación. Las áreas de proceso, con sus metas y prácticas son:

Planificación del proyecto (PP)

El propósito de la Planificación del Proyecto (PP) es establecer y mantener planes que definan las actividades del proyecto. Esta área de proceso implica las siguientes actividades:

- Desarrollar el plan de proyecto.
- Interactuar de forma apropiada con las partes interesadas relevantes.
- Obtener el compromiso con el plan.
- Mantener el plan.

Las metas y prácticas específicas del área son:

SG 1 Establecer estimaciones

SP 1.1 Estimar el alcance del proyecto

SP 1.2 Establecer las estimaciones de los atributos de los productos de trabajo y de las tareas

SP 1.3 Definir las fases del ciclo de vida del proyecto

SP 1.4 Estimar el esfuerzo y el coste

SG 2 Desarrollar un plan de proyecto

SP 2.1 Establecer el presupuesto y el calendario

SP 2.2 Identificar los riesgos del proyecto

SP 2.3 Planificar la gestión de los datos

SP 2.4 Planificar los recursos del proyecto

SP 2.5 Planificar el conocimiento y las habilidades necesarias

SP 2.6 Planificar la involucración de las partes interesadas

SP 2.7 Establecer el plan del proyecto

SG 3 Obtener el compromiso con el plan

SP 3.1 Revisar los planes que afectan al proyecto

SP 3.2 Conciliar los niveles de trabajo y de recursos

SP 3.3 Obtener el compromiso con el plan

Gestión de requisitos (REQM)

El propósito de la Gestión de Requisitos (REQM) es gestionar los requisitos de los productos y los componentes de producto del proyecto, y asegurar la alineación entre esos requisitos, y los planes y los productos de trabajo del proyecto.

Las metas y práctica específicas del área son:

SG 1 Gestionar los requisitos

SP 1.1 Comprender los requisitos

SP 1.2 Obtener el compromiso sobre los requisitos

SP 1.3 Gestionar los cambios a los requisitos

SP 1.4 Mantener la trazabilidad bidireccional de los requisitos

SP 1.5 Asegurar el alineamiento entre el trabajo del proyecto y los requisitos

Gestión de la configuración (CM)

El propósito de la Gestión de Configuración (CM) es establecer y mantener la integridad de los productos de trabajo utilizando la identificación de la configuración, el control de la configuración, el informe del estado de la configuración y las auditorías de la configuración. Esta área de proceso implica las siguientes actividades:

- Identificar la configuración de los productos de trabajo seleccionados que componen las líneas base en puntos determinados en el tiempo.
- Controlar los cambios a los elementos de configuración.
- Construir o proporcionar las especificaciones para construir los productos de trabajo a partir del sistema de gestión de configuración.
- Mantener la integridad de las líneas base.
- Proporcionar a los desarrolladores, usuarios finales y clientes datos precisos del estado y de la configuración actual.

Las metas y práctica específicas del área son:

SG 1 Establecer las líneas base

SP 1.1 Identificar los elementos de configuración.

SP 1.2 Establecer un sistema de gestión de configuración

SP 1.3 Crear o liberar las líneas base

SG 2 Seguir y controlar los cambios

SP 2.1 Seguir las peticiones de cambio

SP 2.2 Controlar los elementos de configuración

SG 3 Establecer la integridad

SP 3.1 Establecer los registros de gestión de configuración

SP 3.2 Realizar auditorías de configuración

2.3.2 Planificación de la implementación

El objetivo del plan es adoptar, adaptar, o vetar las prácticas CMMI de las áreas de proceso: Planificación del proyecto (PP), Gestión de requisitos (REQM), y Gestión de la configuración (CM), para el área de pruebas.

El plan se divide en 3 fases: Capacitación, Implementación, y Cierre.

Capacitación

Su objetivo es que todos los miembros del equipo de pruebas obtengan los conocimientos básicos de CMMI, y en especial de las áreas de proceso que van a ser implementadas.

Implementación

Su objetivo es implementar las prácticas del área de proceso actual. Es decir, esta fase se repetirá por cada área de proceso. La fase consta de las siguientes etapas:

1. Evaluación de la práctica específica: Es aquí donde el equipo evaluará la práctica específica, y hará las siguientes preguntas:
 - ¿Es relevante para el equipo de pruebas?
 - ¿Se puede soportar la práctica con algún proceso ya existente?
 - ¿Se requiere un nuevo proceso para implementar la práctica?
2. Plan de implementación de la práctica: Aquí se define como se va a implementar la práctica.
 - ¿Cuales son los pasos para aplicar la práctica?
 - ¿Cuales son los recursos para aplicar la práctica?
 - Documentación de la implementación de la práctica.
3. Definición de métricas: Se define si se requieren o no métricas para la posterior evaluación de la práctica.
 - ¿Se requieren métricas para la adopción de la práctica?
 - ¿Cuales son las métricas que permitirán evaluar la correcta adopción de la práctica?
 - ¿Cuanto tiempo se evaluará la métrica?
4. Implementación: Aquí se ejecutará el plan definido previamente.

Cierre

Esta fase tiene como objetivo la elaboración de un informe a gerencia sobre el proceso de implementación de las áreas de proceso seleccionadas.

2.3.3 Plan de implementación

La Figura 2.3, muestra una parte de la ejecución del plan en caso de que éste se iniciase el 30 de septiembre de 2019. El plan se encuentra documentado, y se utilizó Microsoft Project para el caso. Finalmente, el plan completo se encuentra en el anexo 4.

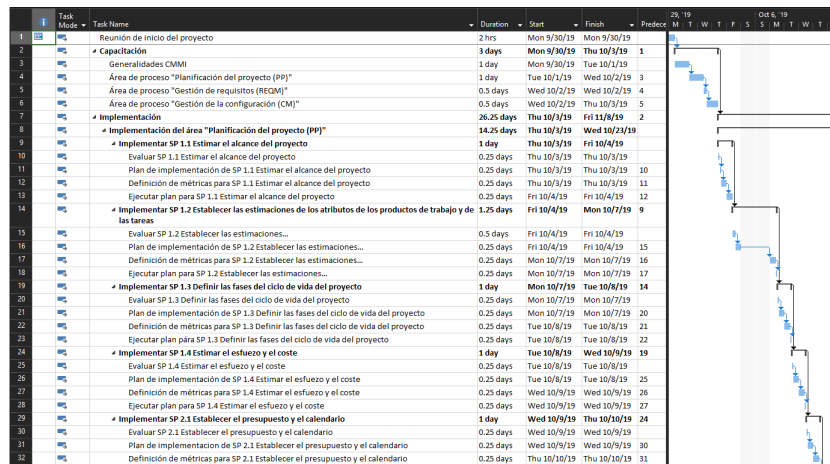


Figura 2.3: Muestra del plan implementación.

2.3.4 Métricas

En el siguiente apartado se considera la métrica que es la razón de ser de este estudio, ya que el objetivo final que es que se mitiguen las apariciones de casos críticos reportados por clientes. Adicionalmente, se propone otra métrica dirigida al mejoramiento del descubrimiento de defectos en otras fases del ciclo de vida del desarrollo de aplicaciones al interior de la empresa por parte del equipo de pruebas. En base a esto se tienen:

1. El número de casos críticos reportados por clientes: Ya que existen varios motivos por los que un caso crítico puede darse, como se puede ver en la Tabla 3.2. Se busca que los casos críticos reportados que tengan como motivo: pruebas de regresión, casos de pruebas incompletos o faltantes, manejo de requerimientos, y administración de la configuración, disminuyan después de la implementación de estas prácticas. Aunque es una métrica que afecta prácticamente a las áreas de desarrollo, análisis de negocios y TI. El equipo de pruebas contribuiría a que los casos críticos disminuyan.
2. El número de defectos (bugs) encontrados por el equipo de los analistas de negocios: Ésta muestra los defectos que el equipo de pruebas no encontró durante las fases de desarrollo regulares, y que a su vez fueron identificados por el equipo de analistas. Finalmente lo que se quiere es mejorar la efectividad y la eficiencia del equipo de pruebas en encontrar defectos, ya que un caso crítico reportado por un cliente no es más que un defecto que no se pudo encontrar durante las fases de desarrollo actuales.

A estas dos métricas se sumaran las que sean definidas por el equipo durante la implementación de las prácticas seleccionadas.

Capítulo 3

Análisis de resultados

El capítulo actual describe los resultados obtenidos basados en el análisis de la base de datos alimentada con la metadata de los casos críticos (resumidos en la Tabla A.1).

3.1 Áreas o procesos fallidos

La ejecución del método causa-raíz descrito en las secciones 2.1.1 y 2.2.1 permitió analizar y agrupar la información en patrones. Estos patrones se asocian con los procedimientos existentes dentro de la empresa, los mismos que fallaron en evidenciar el defecto antes de la “puesta en producción” de un proyecto o mantenimiento programado. Es decir, el caso crítico y su resolución son la consecuencia de fallas en procedimientos al interior de la empresa, que sucedieron durante el desarrollo de proyectos. Estos procedimientos y patrones se encuentran listados en la Tabla 3.1.

Los casos críticos analizados tuvieron en varias ocasiones más de un proceso fallido asociado y, al realizar una agrupación de los casos por procedimiento fallido se obtuvieron los resultados descritos en la Tabla 3.2.

Al buscar información agrupada por caso crítico se encontró que varios casos fueron en resultado de la falla de más de un procedimiento, dando como resultado los datos de la Tabla 3.3.

3.2 Costos asociados a los casos críticos

El presente estudio muestra como se asocian los costos del grupo de recursos con las actividades identificadas. Adicionalmente, los valores han sido modificados para proteger los datos de la empresa.

Procedimiento	Patrón
System Analysis	El análisis del sistema que se realizó como parte del desarrollo original fue incompleto o nulo.
Regression Test	Las pruebas de regresión fueron insuficientes.
Test cases	Los casos de pruebas fueron insuficientes.
Requirements	El requerimiento estuvo incompleto o confuso.
Code change management	No se llevó a cabo un control de cambios apropiado en el código fuente.
Urgent requirement	Cambio urgente por petición del cliente.
known Issue	Error preexistente y conocido.
Wrong prioritization	Priorización incorrecta del requerimiento.
Environment	El entorno de pruebas no estuvo configurado apropiadamente.

Tabla 3.1: Procedimientos fallidos.

Grupo de recursos

Al ser una empresa con su propio equipo de desarrollo de software, los recursos identificados son los roles que están involucrados en el manejo y resolución de los casos críticos, la recopilación de información arrojó la siguiente lista de roles: Personal de soporte al cliente, Analista de negocios, Programador, Tester, personal de paso a producción. Los costos anuales asociados a cada rol se muestra en la Tabla 3.4.

Actividades

Las actividades identificadas, como se explicó en la sección 2.2.2 son el resultado de agrupar tareas por rol. Para obtener la lista de tareas se solicitó al personal involucrado que liste las tareas que ejecutan al encargarse un caso crítico. Esta recopilación de información originó la siguiente lista de actividades: Analizar requerimientos, crear y ejecutar tareas operativas, reproducir el error, codificar cambios, ejecutar pruebas técnicas, ejecutar pruebas de aceptación, actualizar repositorios de código, actualizar documentación, e instalar aplicaciones en producción.

El costo de las actividades se puede calcular mediante el porcentaje de uso de la actividad por parte de un recurso. Como por ejemplo, ya que un analista de negocios ocupó un 2.26 % de su tiempo anual en actividades relacionadas con la resolución de caos críticos, entonces 2.26 % de su salario anual es asociado con estas actividades. Los valores recopilados se pueden revisar en la Tabla 3.5.

El tiempo que los diferentes roles ocuparon en la resolución de los casos críticos se obtuvo

Procedimiento	Número de casos en el que falló
System Analysis	16
Regression Test	12
Test cases	10
Requirements	5
Urgent requiremement	3
known Issue	2
Code change management	1
Environment	1
Wrong prioritization	1

Tabla 3.2: Casos críticos por procedimiento.

Número de casos	Número de procedimientos fallidos asociados
4	3
12	2
15	1

Tabla 3.3: Procedimientos fallidos recurrentes.

de las tareas registradas en el sistema SDLC.

Personal(Recurso)	Costo Anual	Total Dias-Hombre	Costo por Día-Hombre
Soporte al cliente	\$215,000.00	1095	\$196.35
Analista de negocios	\$294,000.00	1460	\$201.37
Programador	\$349,050.00	5110	\$68.31
Tester	\$77,350.00	1095	\$70.46
Personal paso producción	\$91,650.00	1095	\$83.70
Total]	\$1,027,050.00	9855	\$620.36

Tabla 3.4: Costos de los grupos de recursos.




























Recurso	% de tiempo utilizado	Costo anual	Dias-Hombre utilizados	Costo en resolución de casos críticos
Soporte al cliente	%0.50	\$215,000.00	5.5	\$1,079.91
Analista de negocios	%2.26	\$294,000.00	33	\$6,645.21
Programador	%0.91	\$349,050.00	46.5	\$3,176.29
Tester	%2.92	\$77,350.00	32	\$2,260.46
Personal paso producción	%7.12	\$91,650.00	78	\$6,528.49
Total por día	%13.72		195	\$19,690.35

































Tabla 3.5: Costos de las actividades.

































Capítulo 4




Plan de Implementación



















El plan de Implementación fué generado en Microsoft Project, y se adjunto a continuación.

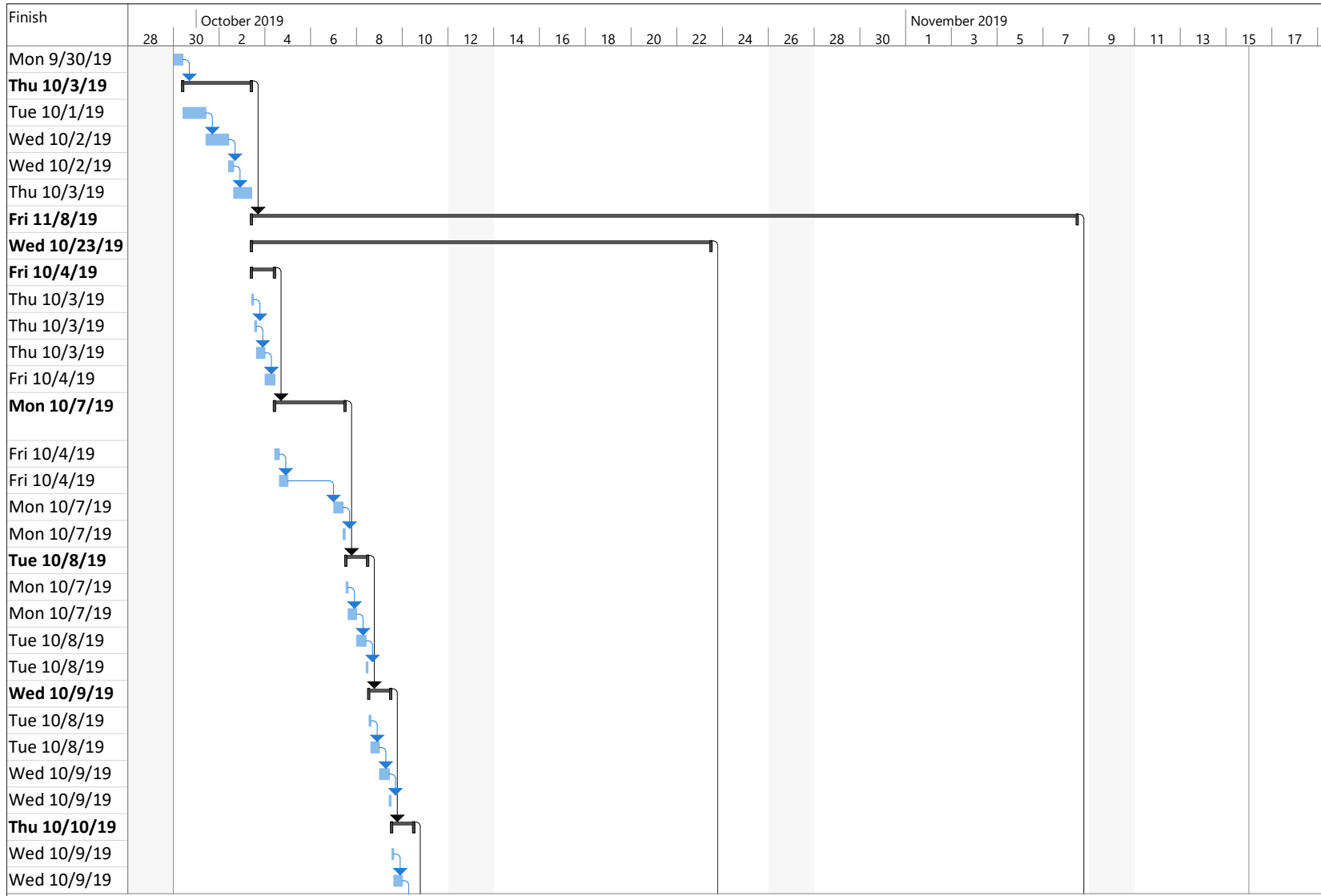
ID	 Task Mode	Task Name	Duration	Start
1		Reunión de inicio del proyecto	2 hrs	Mon 9/30/19
2		Capacitación	3 days	Mon 9/30/19
3		Generalidades CMMI	1 day	Mon 9/30/19
4		Área de proceso "Planificación del proyecto (PP)"	1 day	Tue 10/1/19
5		Área de proceso "Gestión de requisitos (REQM)"	0.5 days	Wed 10/2/19
6		Área de proceso "Gestión de la configuración (CM)"	0.5 days	Wed 10/2/19
7		Implementación	26.25 days	Thu 10/3/19
8		Implementación del área "Planificación del proyecto (PP)"	14.25 days	Thu 10/3/19
9		Implementar SP 1.1 Estimar el alcance del proyecto	1 day	Thu 10/3/19
10		Evaluar SP 1.1 Estimar el alcance del proyecto	0.25 days	Thu 10/3/19
11		Plan de implementación de SP 1.1 Estimar el alcance del proyecto	0.25 days	Thu 10/3/19
12		Definición de métricas para SP 1.1 Estimar el alcance del proyecto	0.25 days	Thu 10/3/19
13		Ejecutar plan para SP 1.1 Estimar el alcance del proyecto	0.25 days	Fri 10/4/19
14		Implementar SP 1.2 Establecer las estimaciones de los atributos de los productos de trabajo y de las tareas	1.25 days	Fri 10/4/19
15		Evaluar SP 1.2 Establecer las estimaciones...	0.5 days	Fri 10/4/19
16		Plan de implementación de SP 1.2 Establecer las estimaciones...	0.25 days	Fri 10/4/19
17		Definición de métricas para SP 1.2 Establecer las estimaciones...	0.25 days	Mon 10/7/19
18		Ejecutar plan para SP 1.2 Establecer las estimaciones...	0.25 days	Mon 10/7/19
19		Implementar SP 1.3 Definir las fases del ciclo de vida del proyecto	1 day	Mon 10/7/19
20		Evaluar SP 1.3 Definir las fases del ciclo de vida del proyecto	0.25 days	Mon 10/7/19
21		Plan de implementación de SP 1.3 Definir las fases del ciclo de vida del proyecto	0.25 days	Mon 10/7/19
22		Definición de métricas para SP 1.3 Definir las fases del ciclo de vida del proyecto	0.25 days	Tue 10/8/19
23		Ejecutar plan para SP 1.3 Definir las fases del ciclo de vida del proyecto	0.25 days	Tue 10/8/19
24		Implementar SP 1.4 Estimar el esfuerzo y el coste	1 day	Tue 10/8/19
25		Evaluar SP 1.4 Estimar el esfuerzo y el coste	0.25 days	Tue 10/8/19
26		Plan de implementación de SP 1.4 Estimar el esfuerzo y el coste	0.25 days	Tue 10/8/19
27		Definición de métricas para SP 1.4 Estimar el esfuerzo y el coste	0.25 days	Wed 10/9/19
28		Ejecutar plan para SP 1.4 Estimar el esfuerzo y el coste	0.25 days	Wed 10/9/19
29		Implementar SP 2.1 Establecer el presupuesto y el calendario	1 day	Wed 10/9/19
30		Evaluar SP 2.1 Establecer el presupuesto y el calendario	0.25 days	Wed 10/9/19
31		Plan de implementación de SP 2.1 Establecer el presupuesto y el calendario	0.25 days	Wed 10/9/19

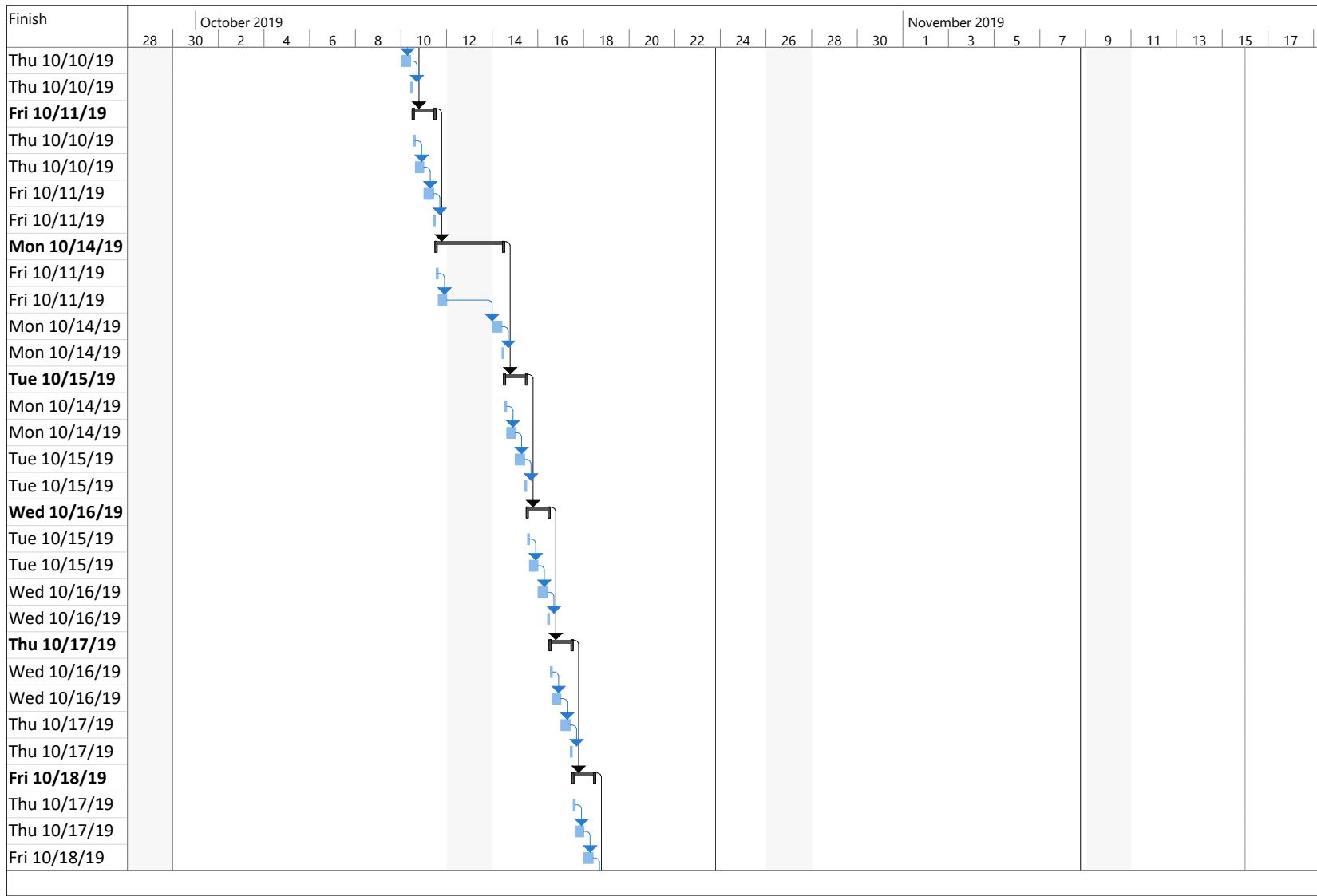
ID	 Task Mode	Task Name	Duration	Start
32		Definición de métricas para SP 2.1 Establecer el presupuesto y el calendario	0.25 days	Thu 10/10/19
33		Ejecutar plan para SP 2.1 Establecer el presupuesto y el calendario	0.25 days	Thu 10/10/19
34		Implementar SP 2.2 Identificar los riesgos del proyecto	1 day	Thu 10/10/19
35		Evaluar SP 2.2 Identificar los riesgos del proyecto	0.25 days	Thu 10/10/19
36		Plan de implementación de SP 2.2 Identificar los riesgos del proyecto	0.25 days	Thu 10/10/19
37		Definición de métricas para SP 2.2 Identificar los riesgos del proyecto	0.25 days	Fri 10/11/19
38		Ejecutar plan para SP 2.2 Identificar los riesgos del proyecto	0.25 days	Fri 10/11/19
39		Implementar SP 2.3 Planificar la gestión de los datos	1 day	Fri 10/11/19
40		Evaluar SP 2.3 Planificar la gestión de los datos	0.25 days	Fri 10/11/19
41		Plan de implementación de SP 2.3 Planificar la gestión de los datos	0.25 days	Fri 10/11/19
42		Definición de métricas para SP 2.3 Planificar la gestión de los datos	0.25 days	Mon 10/14/19
43		Ejecutar plan para SP 2.3 Planificar la gestión de los datos	0.25 days	Mon 10/14/19
44		Implementar SP 2.4 Planificar los recursos del proyecto	1 day	Mon 10/14/19
45		Evaluar SP 2.4 Planificar los recursos del proyecto	0.25 days	Mon 10/14/19
46		Plan de implementación de SP 2.4 Planificar los recursos del proyecto	0.25 days	Mon 10/14/19
47		Definición de métricas para SP 2.4 Planificar los recursos del proyecto	0.25 days	Tue 10/15/19
48		Ejecutar plan para SP 2.4 Planificar los recursos del proyecto	0.25 days	Tue 10/15/19
49		Implementar SP 2.5 Planificar el conocimiento y las habilidades necesarias	1 day	Tue 10/15/19
50		Evaluar SP 2.5 Planificar el conocimiento y las habilidades necesarias	0.25 days	Tue 10/15/19
51		Plan de implementación de SP 2.5 Planificar el conocimiento y las ...	0.25 days	Tue 10/15/19
52		Definición de métricas para SP 2.5 Planificar el conocimiento y las ...	0.25 days	Wed 10/16/19
53		Ejecutar plan para SP 2.5 Planificar el conocimiento y las habilidades necesarias	0.25 days	Wed 10/16/19
54		Implementar SP 2.6 Planificar la involucración de las partes interesadas	1 day	Wed 10/16/19
55		Evaluar SP 2.6 Planificar la involucración de las partes interesadas	0.25 days	Wed 10/16/19
56		Plan de implementación de SP 2.6 Planificar la involucración de las ...	0.25 days	Wed 10/16/19
57		Definición de métricas para SP 2.6 Planificar la involucración de las ...	0.25 days	Thu 10/17/19
58		Ejecutar plan para SP 2.6 Planificar la involucración de las partes interesadas	0.25 days	Thu 10/17/19
59		Implementar SP 2.7 Establecer el plan del proyecto	1 day	Thu 10/17/19
60		Evaluar SP 2.7 Establecer el plan del proyecto	0.25 days	Thu 10/17/19
61		Plan de implementación de SP 2.7 Establecer el plan del proyecto	0.25 days	Thu 10/17/19
62		Definición de métricas para SP 2.7 Establecer el plan del proyecto	0.25 days	Fri 10/18/19

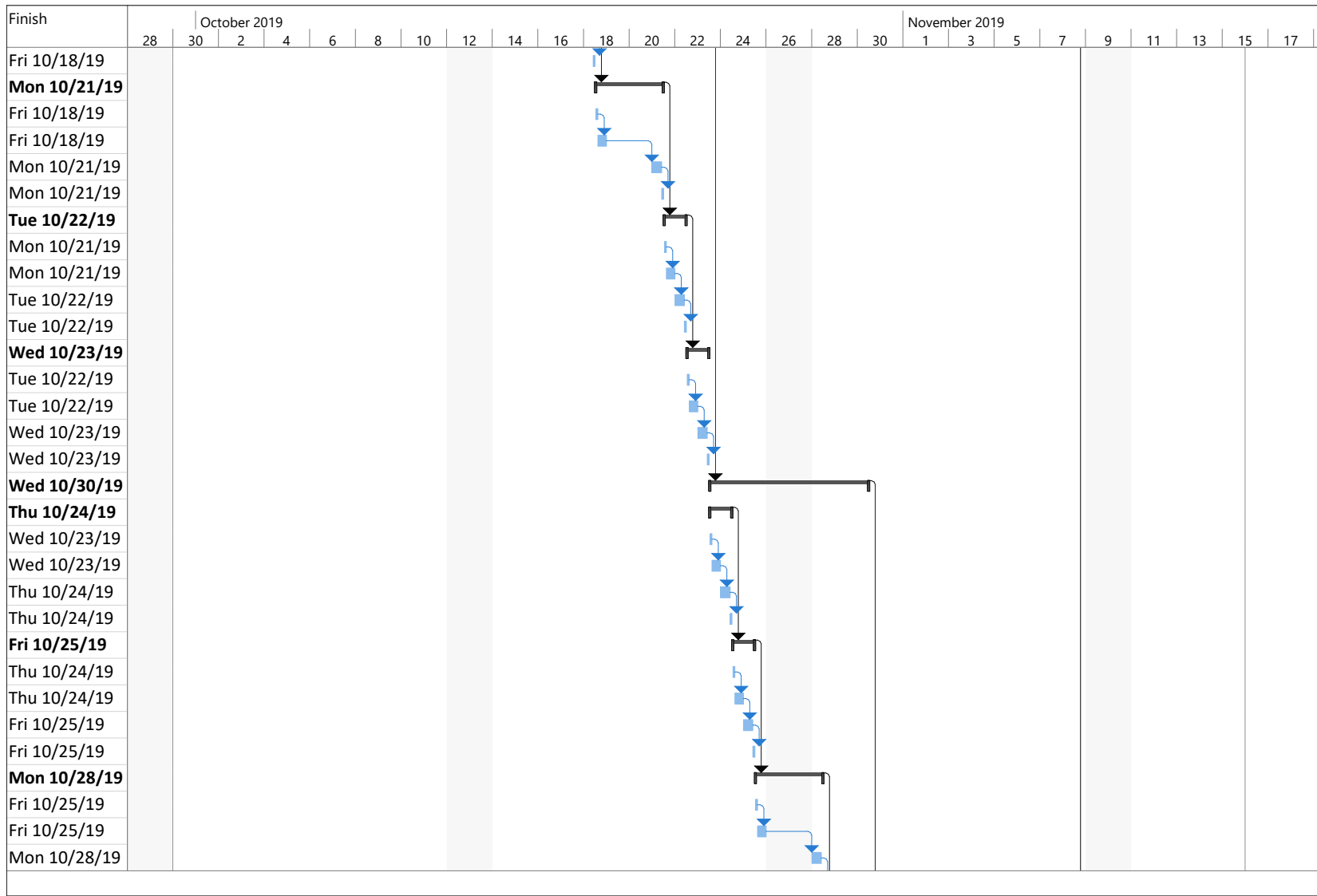
ID	 Task Mode	Task Name	Duration	Start
63		Ejecutar plan para SP 2.7 Establecer el plan del proyecto	0.25 days	Fri 10/18/19
64		Implementar SP 3.1 Revisar los planes que afectan al proyecto	1 day	Fri 10/18/19
65		Evaluar SP 3.1 Revisar los planes que afectan al proyecto	0.25 days	Fri 10/18/19
66		Plan de implementación para SP 3.1 Revisar los planes que afectan al proyecto	0.25 days	Fri 10/18/19
67		Definición de métricas para SP 3.1 Revisar los planes que afectan al proyecto	0.25 days	Mon 10/21/19
68		Ejecutar plan para SP 3.1 Revisar los planes que afectan al proyecto	0.25 days	Mon 10/21/19
69		Implementar SP 3.2 Conciliar los niveles de trabajo y de recursos	1 day	Mon 10/21/19
70		Evaluar SP 3.2 Conciliar los niveles de trabajo y de recursos	0.25 days	Mon 10/21/19
71		Plan de implementación SP 3.2 Conciliar los niveles de trabajo y de recursos	0.25 days	Mon 10/21/19
72		Definición de métricas para SP 3.2 Conciliar los niveles de trabajo y de recursos	0.25 days	Tue 10/22/19
73		Ejecutar plan para SP 3.2 Conciliar los niveles de trabajo y de recursos	0.25 days	Tue 10/22/19
74		Implementar SP 3.3 Obtener el compromiso con el plan	1 day	Tue 10/22/19
75		Evaluar SP 3.3 Obtener el compromiso con el plan	0.25 days	Tue 10/22/19
76		Plan de implementación para SP 3.3 Obtener el compromiso con el plan	0.25 days	Tue 10/22/19
77		Definición de métricas para SP 3.3 Obtener el compromiso con el plan	0.25 days	Wed 10/23/19
78		Ejecutar plan para SP 3.3 Obtener el compromiso con el plan	0.25 days	Wed 10/23/19
79		Implementación del área "Gestión de requisitos (REQM)"	5 days	Wed 10/23/19
80		Implementar SP 1.1 Comprender los requisitos	1 day	Wed 10/23/19
81		Evaluar SP 1.1 Comprender los requisitos	0.25 days	Wed 10/23/19
82		Plan de implementación para SP 1.1 Comprender los requisitos	0.25 days	Wed 10/23/19
83		Definición de métricas para SP 1.1 Comprender los requisitos	0.25 days	Thu 10/24/19
84		Ejecutar plan para SP 1.1 Comprender los requisitos	0.25 days	Thu 10/24/19
85		Implementar SP 1.2 Obtener el compromiso sobre los requisitos	1 day	Thu 10/24/19
86		Evaluar SP 1.2 Obtener el compromiso sobre los requisitos	0.25 days	Thu 10/24/19
87		Plan de implementación para SP 1.2 Obtener el compromiso sobre los requisitos	0.25 days	Thu 10/24/19
88		Definición de métricas para SP 1.2 Obtener el compromiso sobre los requisitos	0.25 days	Fri 10/25/19
89		Ejecutar plan para SP 1.2 Obtener el compromiso sobre los requisitos	0.25 days	Fri 10/25/19
90		Implementar SP 1.3 Gestionar los cambios a los requisitos	1 day	Fri 10/25/19
91		Evaluar SP 1.3 Gestionar los cambios a los requisitos	0.25 days	Fri 10/25/19
92		Plan de implementación para SP 1.3 Gestionar los cambios a los requisitos	0.25 days	Fri 10/25/19
93		Definición de métricas para SP 1.3 Gestionar los cambios a los requisitos	0.25 days	Mon 10/28/19

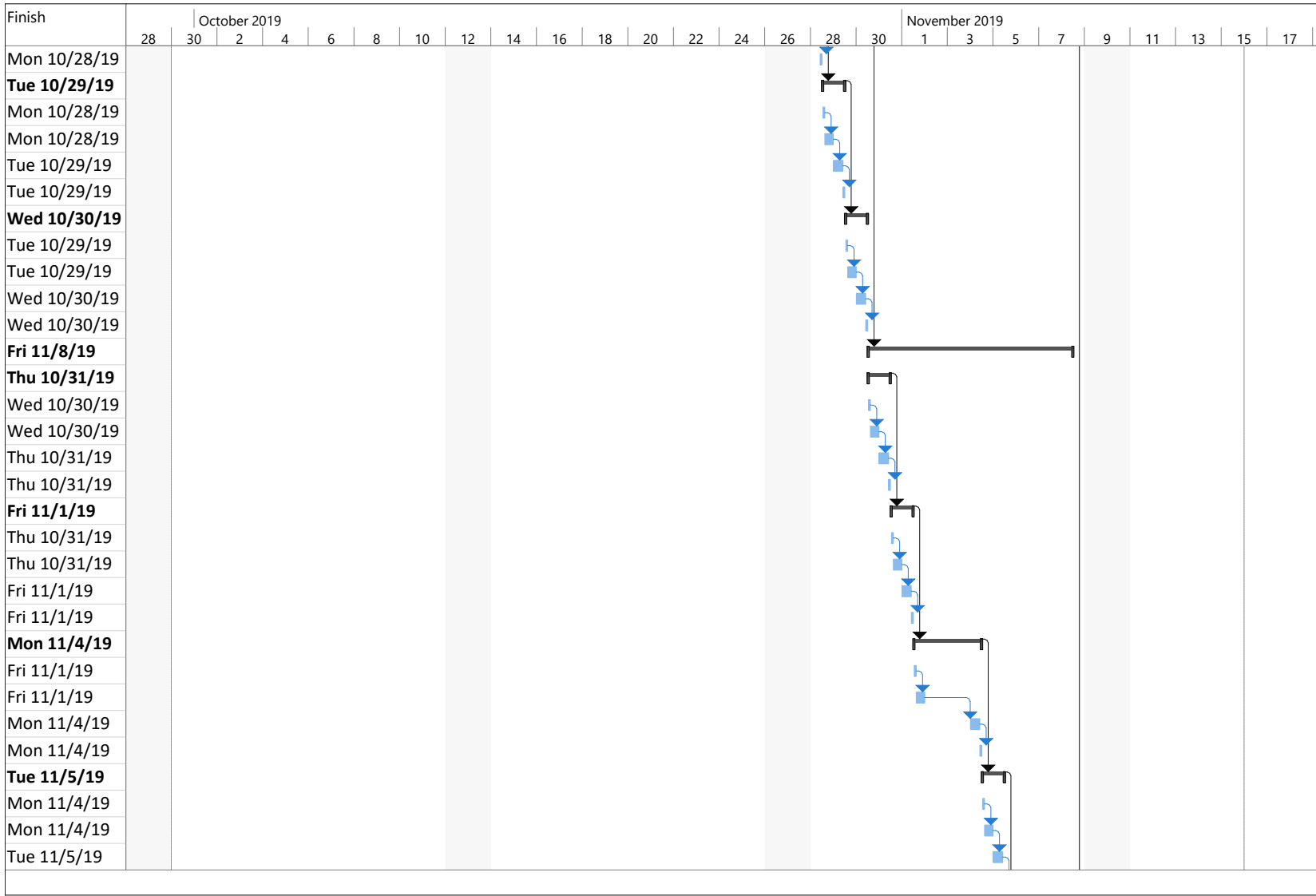
ID	 Task Mode	Task Name	Duration	Start
94		Ejecutar plan para SP 1.3 Gestionar los cambios a los requisitos	0.25 days	Mon 10/28/19
95		Implementar SP 1.4 Mantener la trazabilidad bidireccional de los requisitos	1 day	Mon 10/28/19
96		Evaluar SP 1.4 Mantener la trazabilidad bidireccional de los requisitos	0.25 days	Mon 10/28/19
97		Plan de implementación para SP 1.4 Mantener la trazabilidad bidireccional ...	0.25 days	Mon 10/28/19
98		Definición de métricas para SP 1.4 Mantener la trazabilidad bidireccional de los requisitos	0.25 days	Tue 10/29/19
99		Ejecutar plan para SP 1.4 Mantener la trazabilidad bidireccional de los requisitos	0.25 days	Tue 10/29/19
100		Implementar SP 1.5 Asegurar el alineamiento entre el trabajo del proyecto y los requisitos	1 day	Tue 10/29/19
101		Evaluación de SP 1.5 Asegurar el alineamiento entre el trabajo del proyecto y ...	0.25 days	Tue 10/29/19
102		Plan de implementación de SP 1.5 Asegurar el alineamiento entre el trabajo del proy ...	0.25 days	Tue 10/29/19
103		Definición de métricas para SP 1.5 Asegurar el alineamiento entre el trabajo del proy ...	0.25 days	Wed 10/30/19
104		Ejecutar plan para SP 1.5 Asegurar el alineamiento entre el trabajo del proyecto y ...	0.25 days	Wed 10/30/19
105		Implementación del área "Gestión de la configuración (CM)"	7 days	Wed 10/30/19
106		Implementar SP 1.1 Identificar los elementos de configuración	1 day	Wed 10/30/19
107		Evaluación de SP 1.1 Identificar los elementos de configuración	0.25 days	Wed 10/30/19
108		Plan de implementación de SP 1.1 Identificar los elementos de configuración	0.25 days	Wed 10/30/19
109		Definición de métricas para SP 1.1 Identificar los elementos de configuración	0.25 days	Thu 10/31/19
110		Ejecutar plan para SP 1.1 Identificar los elementos de configuración	0.25 days	Thu 10/31/19
111		Implementar SP 1.2 Establecer un sistema de gestión de configuración	1 day	Thu 10/31/19
112		Evaluación SP 1.2 Establecer un sistema de gestión de configuración	0.25 days	Thu 10/31/19
113		Plan de implementación de SP 1.2 Establecer un sistema de gestión de configuración	0.25 days	Thu 10/31/19
114		Definición de métricas para SP 1.2 Establecer un sistema de gestión de configuración	0.25 days	Fri 11/1/19
115		Ejecutar plan para SP 1.2 Establecer un sistema de gestión de configuración	0.25 days	Fri 11/1/19
116		Implementar SP 1.3 Crear o liberar las líneas base	1 day	Fri 11/1/19
117		Evaluar SP 1.3 Crear o liberar las líneas base	0.25 days	Fri 11/1/19
118		Plan de implementación de SP 1.3 Crear o liberar las líneas base	0.25 days	Fri 11/1/19
119		Definición de métricas para SP 1.3 Crear o liberar las líneas base	0.25 days	Mon 11/4/19
120		Ejecutar plan para SP 1.3 Crear o liberar las líneas base	0.25 days	Mon 11/4/19
121		Implementar SP 2.1 Seguir las peticiones de cambio	1 day	Mon 11/4/19
122		Evaluar SP 2.1 Seguir las peticiones de cambio	0.25 days	Mon 11/4/19
123		Plan de implementación de SP 2.1 Seguir las peticiones de cambio	0.25 days	Mon 11/4/19
124		Definición de métricas para SP 2.1 Seguir las peticiones de cambio	0.25 days	Tue 11/5/19

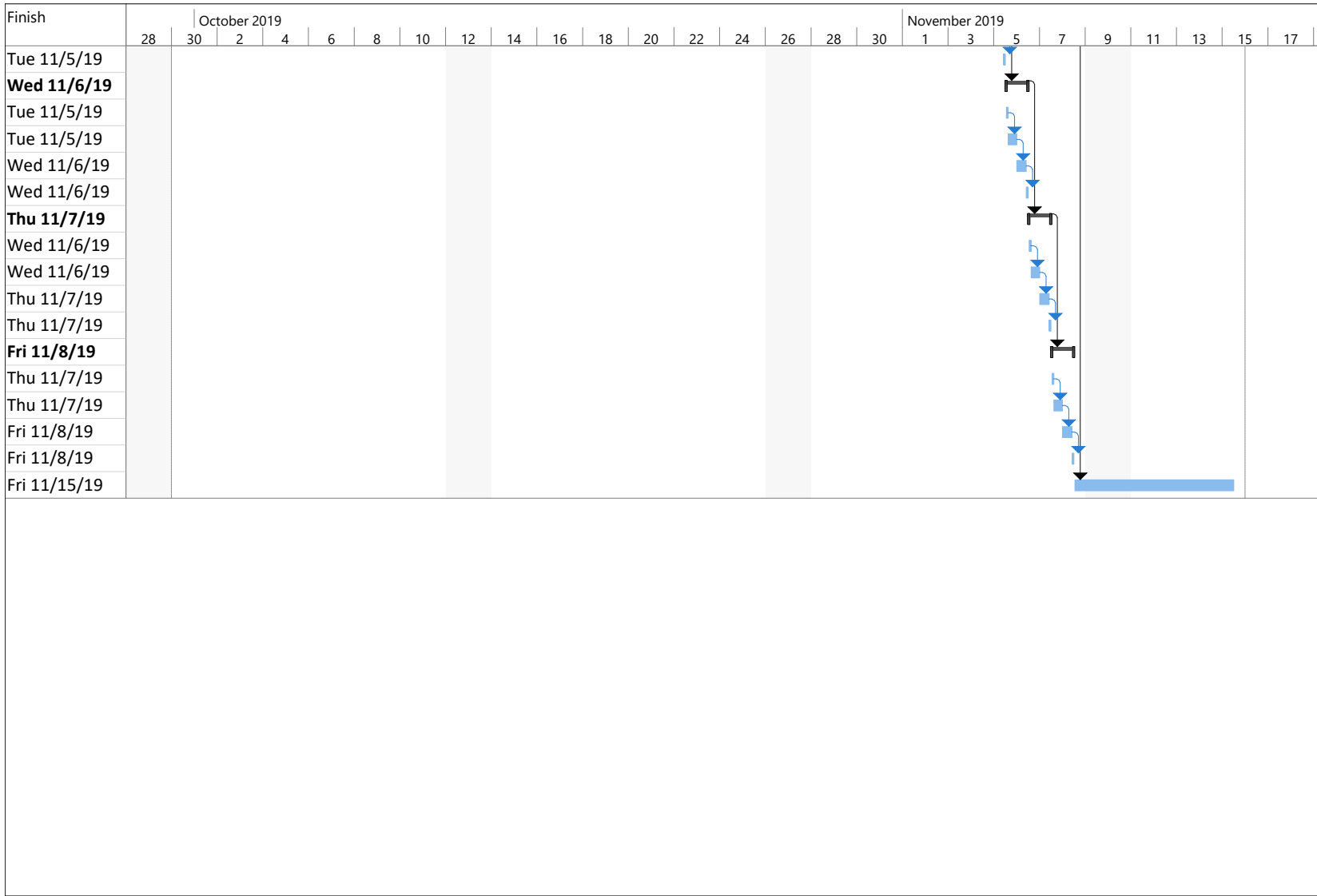
ID	 Task Mode	Task Name	Duration	Start
125		Ejecutar plan para SP 2.1 Seguir las peticiones de cambio	0.25 days	Tue 11/5/19
126		Implementar SP 2.2 Controlar los elementos de configuración	1 day	Tue 11/5/19
127		Evaluar SP 2.2 Controlar los elementos de configuración	0.25 days	Tue 11/5/19
128		Plan de implementación para SP 2.2 Controlar los elementos de configuración	0.25 days	Tue 11/5/19
129		Definición de métricas para SP 2.2 Controlar los elementos de configuración	0.25 days	Wed 11/6/19
130		Ejecutar plan para SP 2.2 Controlar los elementos de configuración	0.25 days	Wed 11/6/19
131		Implementar SP 3.1 Establecer los registros de gestión de configuración	1 day	Wed 11/6/19
132		Evaluar SP 3.1 Establecer los registros de gestión de configuración	0.25 days	Wed 11/6/19
133		Plan de implementación para SP 3.1 Establecer los registros de gestión de configuración	0.25 days	Wed 11/6/19
134		Definición de métricas para SP 3.1 Establecer los registros de gestión de configuración	0.25 days	Thu 11/7/19
135		Ejecutar plan para SP 3.1 Establecer los registros de gestión de configuración	0.25 days	Thu 11/7/19
136		Implementar SP 3.2 Realizar auditorías de configuración	1 day	Thu 11/7/19
137		Evaluar SP 3.2 Realizar auditorías de configuración	0.25 days	Thu 11/7/19
138		Plan de implementación para SP 3.2 Realizar auditorías de configuración	0.25 days	Thu 11/7/19
139		Definición de métricas para SP 3.2 Realizar auditorías de configuración	0.25 days	Fri 11/8/19
140		Ejecutar el plan para SP 3.2 Realizar auditorías de configuración	0.25 days	Fri 11/8/19
141		Informe Final	5 days	Fri 11/8/19













Project: Implementacion Date: Thu 9/5/19	Task		Inactive Summary		External Tasks	
	Split		Manual Task		External Milestone	
	Milestone		Duration-only		Deadline	
	Summary		Manual Summary Rollup		Progress	
	Project Summary		Manual Summary		Manual Progress	
	Inactive Task		Start-only			
	Inactive Milestone		Finish-only			

Capítulo 5

Conclusiones y Recomendaciones

5.1 Conclusiones

1. El porcentaje propuesto inicialmente no se pudo mantener durante el estudio debido a que la diversidad de los casos criticos no representaban consistentemente al universo de casos para el año 2017. Por lo que se pasó de un enfoque probabilistico, a uno no probabilistico.
2. El análisis de costos muestra a lo que la empresa se encuentra expuesta si es que no toma acción para la prevención de casos críticos.
3. La identificación de patrones demostró ser un proceso largo debido a la documentación incompleta y a la búsqueda de evidencias que soporten al mismo.
4. La identificación de patrones permitió identificar las áreas responsables de los procesos fallidos de una manera transparente.
5. La definición de métricas deberá ser un producto de la implementación de los procesos propuestos, ya que estas deben responder a las necesidades de la empresa.
6. La identificación de patrones también ayudó a encontra las áreas de proceso CMMi que mejor se acoplen a la mitigación de los problemas evidenciados por los casos críticos.
7. No se justifica la implementación de áreas de proceso o prácticas CMMi adicionales a las que el presente trabajo propone, debido a que las áreas de proceso propuestas cubrirían los patrones expuestos por los casos criticos analizados.

5.2 Recomendaciones

1. Los resultados obtenidos por la empresa son la suma de los esfuerzos de las diferentes áreas que la componen, por lo que mejorar los procesos del área de pruebas contribuirá

de manera positiva a los resultados globales. Sin embargo, si se desea maximizar estos resultados, se deberán tomar iniciativas de mejoramiento en la calidad de los procesos y los productos de las demás áreas de la empresa.

2. La selección del área a mejorar va a depender enteramente de los beneficios que la empresa busque. Sin embargo todas las áreas pueden analizar, encontrar, y sobre todo proponer iniciativas que aporten a los resultados globales de la empresa. Esto se puede extender incluso a nivel personal, ya que el trabajo que una persona realiza impacta en mayor o menor grado en toda la organización.
3. Las iniciativas de mejora pueden verse como más trabajo al momento de implementarlas, y ciertamente causaran un impacto en el personal que trabaje en ellas, ya que implican cambios a nivel personal y profesional. Lo que no se ve en ese momento es que el trabajo realizado va a ver reflejado como beneficios en tiempo, esfuerzo, escalabilidad, entre otros. Por lo que los grupos que busquen e implementen exitosamente procesos y productos de calidad se verá mejor posicionados, siendo capaces de crecer como equipo y volverse estratégicos dentro de una empresa.

Apéndice A

Estructura de la base de datos de errores

La base de datos de los errores críticos se compone de las siguientes tablas:

Elemento de base de datos	Propósito
Tabla PasoProduccion	Registra el paso a producción de uno o más casos críticos. Los datos que contiene se detalla en la Tabla A.2.
Tabla CasoCritico	Registra los datos de un caso crítico. Los datos que contiene se detalla en la Tabla A.3.
Tabla Clasificacion	Identifica los patrones que se fueron encontrando durante el análisis de los datos. Los datos que contiene se detalla en la Tabla A.4.
Tabla ClasificacionCasoCritico	Permite romper la relación muchos-a-muchos entre las tablas de CasoCritico y Clasificacion. Los datos que contiene se detalla en la Tabla A.5.
Elementos de soporte	Se crearon procedimientos almacenados y funciones.

Tabla A.1: Elementos de la base de datos DataCasosCriticos.

Columna	Propósito
pp_id	Es el Id del paso a producción, que tiene una codificación definida por la empresa. La cual es (Año)+(número de paso a producción en el año)+(CR para indicar si es crítico). Ejemplo: 20170001CR.
pp_feature	Es el identificador del paso a producción dentro del sistema SDLC. Esto permite ubicar el paso a producción en este otro sistema.
pp_ArchivosAfectados	Es la cantidad de archivos que necesitaron ser modificados para resolver el caso crítico.
pp_InicioRelease	Es la fecha en la que se inició con la planificación del paso a producción del o de los casos críticos.
pp_FinRelease	Es la fecha en la que se culminó con la última tarea de la planificación del paso a producción.

Tabla A.2: Datos que almacena la tabla PasoProduccion.

Columna	Propósito
cc_id	Es el identificador del caso crítico dentro del sistema SDLC, es decir se reutilizó el identificador del otro sistema como identificador en la base de datos. Esto permite ubicar el caso crítico en este otro sistema.
pp_id	Es el identificador del paso a producción. Permite enlazar un caso crítico con un paso a producción.
cc.FechaCreacion	Fecha de la creación del caso crítico dentro del sistema SDLC.
cc.FechaCierre	Fecha en la que el caso crítico pasó todas las certificaciones dentro del sistema SDLC.
cc.TiempoSoporte	Tiempo de la resolución del caso desde el punto de vista del equipo de soporte.
cc.CausaUno	Causa del caso crítico.
cc.CausaDos	Causa del caso crítico. Durante el desarrollo de la base de datos no se requirió crear una relación "CasoCritico - Causas".
cc.AccionCorrectivaUno	Acción correctiva por parte de la empresa.
cc.AccionCorrectivaDos	Acción correctiva por parte de la empresa. Durante el desarrollo de la base de datos no se requirió crear una relación "CasoCritico - AccionCorrectiva".
cc.PostPasoProduccion	Indica si el caso crítico se encontró luego de un paso a producción de un proyecto, o de un mantenimiento.

Tabla A.3: Datos que almacena la tabla CasoCritico.

Columna	Propósito
cl_id	Es un secuencial que sirve de llave primaria.
cl_Nombre	Es el nombre del patrón identificado.
cl_Descripcion	Es la descripción del patrón

Tabla A.4: Datos que almacena la tabla Clasificacion.

Columna	Propósito
cl_id	Es el identificador de la clasificación en la que se engloba el caso crítico.
cc_id	Es el identificador del caso crítico que encaja en la clasificación.

Tabla A.5: Datos que almacena la tabla ClasificacionCasoCritico.

Bibliografía

- [1] C. P. Team, “Cmmi for development, version 1.3,” Tech. Rep. CMU/SEI-2010-TR-033, 2010. [Online]. Available: <https://cmmiinstitute.com/getattachment/4439387f-28aa-4f3a-8f2b-a0cc5b449e47/attachment.aspx>
- [2] P. H. Feiler and W. S. Humphrey, “Software process development and enactment: concepts and definitions,” in *[1993] Proceedings of the Second International Conference on the Software Process-Continuous Software Process Improvement*, Feb 1993, pp. 28–40.
- [3] G. Rong, H. Zhang, and D. Shao, “Cmmi guided process improvement for devops projects: An exploratory case study,” in *2016 IEEE/ACM International Conference on Software and System Processes (ICSSP)*, May 2016, pp. 76–85.
- [4] M. Kuhrmann, C. Konopka, P. Nellesmann, P. Diebold, and J. Münch, “Software process improvement: Where is the evidence?: Initial findings from a systematic mapping study,” in *Proceedings of the 2015 International Conference on Software and System Process*, ser. ICSSP 2015. New York, NY, USA: ACM, 2015, pp. 107–116. [Online]. Available: <http://doi.acm.org/10.1145/2785592.2785600>
- [5] R. V. Connor and P. Clarke, “Software process reflexivity and business performance: Initial results from an empirical study,” in *Proceedings of the 2015 International Conference on Software and System Process*, ser. ICSSP 2015. New York, NY, USA: ACM, 2015, pp. 142–146. [Online]. Available: <http://doi.acm.org/10.1145/2785592.2785607>
- [6] A. Dyck, R. Penners, and H. Lichter, “Towards definitions for release engineering and devops,” in *2015 IEEE/ACM 3rd International Workshop on Release Engineering*, May 2015, pp. 3–3.
- [7] S. B. Oré and T. S. Feliu, “Lessons Learned and Software Process Improvement,” *Proceedings of the 7th International Conference on Information Communication and Management - ICICM 2017*, no. 2, pp. 12–17, 2017. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=3134383.3134413>
- [8] T. Toroi, A. Raninen, and H. Vainio, “Using functional defect analysis as an input for software process improvement: Initial results,” in *Communications in*

-
- Computer and Information Science*. Springer, 2011, pp. 181–192. [Online]. Available: https://link.springer.com/chapter/10.1007%2F978-3-642-31199-4_16
- [9] R. M. Fontana, V. Meyer, S. Reinehr, and A. Malucelli, “Progressive Outcomes: A framework for maturing in agile software development,” *Journal of Systems and Software*, vol. 102, pp. 88–108, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0164121214002908>
- [10] B. Nikolik, “Software quality assurance economics,” *Information and Software Technology*, vol. 54, no. 11, pp. 1229 – 1238, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S095058491200105X>
- [11] S. Evans, “Electronic health records: Then, now, and in the future,” 2016, [Online; accessed 01-November-2019]. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5171496/#>
- [12] A. A. of Kidney Patients. Understanding cms crownweb patient data management. [Online]. Available: <https://aakp.org/dialysis/understanding-cms-crownweb-patient-data-management/>
- [13] T. Mäkilä, A. Järvi, M. Rönkkö, and J. Nissilä, “How to define software-as-a-service – an empirical study of finnish saas providers,” vol. 51, 06 2010, pp. 115–124. [Online]. Available: https://www.researchgate.net/publication/221159320_How_to_Define_Software-as-a-Service_-_An_Empirical_Study_of_Finnish_SaaS_Providers
- [14] “What is service-level agreement (sla)?” 2019, [Accessed 01-November-2019]. [Online]. Available: <https://searchitchannel.techtarget.com/definition/service-level-agreement>
- [15] “What is crm (customer relationship management)?” 2019, [Accessed 01-November-2019]. [Online]. Available: <https://searchcustomerexperience.techtarget.com/definition/CRM-customer-relationship-management>
- [16] “What is the software development life cycle (sdlc)?” 2019, [Accessed 01-November-2019]. [Online]. Available: <https://searchsoftwarequality.techtarget.com/definition/software-development-life-cycle-SDLC>
- [17] “The principles of the kanban method,” 2019, [Accessed 01-November-2019]. [Online]. Available: <https://web.archive.org/web/20140114161522/http://www.djaa.com/principles-kanban-method>
- [18] S. S. Patel, *SCRUM Y XP DESDE LAS TRINCHERAS*, 2016, vol. 69, no. 6. [Online]. Available: <http://www.proyectalis.com/wp-content/uploads/2008/02/scrum-y-xp-desde-las-trincheras.pdf>

-
- [19] M. Staples, M. Niazi, R. Jeffery, A. Abrahams, P. Byatt, and R. Murphy, “An exploratory study of why organizations do not adopt CMMI,” *Journal of Systems and Software*, vol. 80, no. 6, pp. 883–895, 2007. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0164121206002573>
- [20] Y. Levy and T. J. Ellis, “A Systems Approach to Conduct an Effective Literature Review in Support of Information Systems Research,” vol. 9, 2006. [Online]. Available: <https://goo.gl/4tCNPq>
- [21] S. Arreche, S. Matalonga, and T. S. Feliu, “Tools for defect causal analysis: A systematic literature review,” in *7th Iberian Conference on Information Systems and Technologies (CISTI 2012)*, June 2012, pp. 1–7.
- [22] M. Söylemez and A. Tarhan, “Challenges of software process and product quality improvement: catalyzing defect root-cause investigation by process enactment data analysis,” 2016.
- [23] A. Raninen, T. Toroi, H. Vainio, and J. Ahonen, “Defect data analysis as input for software process improvement,” in *13th Int. Conference on Product-focused Software Process Improvement*. Springer, 2012, pp. 3–16. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-31063-8_2
- [24] M. Kalinowski, D. N. Card, and G. H. Travassos, “Evidence-based guidelines to defect causal analysis,” *IEEE Software*, vol. 29, no. 4, pp. 16–18, July 2012.
- [25] D. N. Card, “Defect-causal analysis drives down error rates,” *IEEE Software*, vol. 10, no. 4, pp. 98–99, July 1993. [Online]. Available: <https://ieeexplore.ieee.org/document/219639>
- [26] M. Kalinowski, E. Mendes, and G. H. Travassos, “An industry ready defect causal analysis approach exploring bayesian networks,” in *Software Quality Model-Based Approaches for Advanced Software and Systems Engineering*. Springer, 2014, pp. 12–33. [Online]. Available: https://doi.org/10.1007/978-3-319-03602-1_2
- [27] M. Kalinowski, P. Curty, A. Paes, A. Ferreira, R. Spinola, D. M. Fernandez, M. Felderer, and S. Wagner, “Supporting defect causal analysis in practice with cross-company data on causes of requirements engineering problems,” in *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP)*, May 2017, pp. 223–232. [Online]. Available: <https://ieeexplore.ieee.org/document/7965446>
- [28] W. Afzal, S. Alone, K. Glocksien, and R. Torkar, “Software test process improvement approaches: A systematic literature review and an industrial case study,” *Journal of Systems and Software*, vol. 111, pp. 1–33, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0164121215001910>

-
- [29] N. N. Tuan and H. Q. Thang, “Combining maturity with agility: lessons learnt from a case study,” in *Proceedings of the Fourth Symposium on Information and Communication Technology*. ACM, 2013, pp. 267–274.
- [30] S. Christa and V. Suma, “An analysis on the significance of ticket analytics and defect analysis from software quality perspective,” in *2016 International Conference on Inventive Computation Technologies (ICICT)*, vol. 1, Aug 2016, pp. 1–5.
- [31] Z. Xu, J. Liu, X. Luo, and T. Zhang, “Cross-version defect prediction via hybrid active learning with kernel principal component analysis,” in *2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, March 2018, pp. 209–220.
- [32] F. M. Favarò, D. W. Jackson, J. H. Saleh, and D. N. Mavris, “Software contributions to aircraft adverse events: Case studies and analyses of recurrent accident patterns and failure mechanisms,” *Reliability Engineering & System Safety*, vol. 113, pp. 131–142, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0951832013000070>
- [33] C. Couto, C. Silva, M. T. Valente, R. Bigonha, and N. Anquetil, “Uncovering causal relationships between software metrics and bugs,” in *2012 16th European Conference on Software Maintenance and Reengineering*, March 2012, pp. 223–232.
- [34] M. Söylemez and A. Tarhan, “Using process enactment data analysis to support orthogonal defect classification for software process improvement,” in *2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement*, Oct 2013, pp. 120–125.
- [35] P. Janczarek and J. Sosnowski, “Investigating software testing and maintenance reports: Case study,” *Information and Software Technology*, vol. 58, pp. 272–288, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0950584914001542>
- [36] R. Yin, *Applications of case study research*. Thousand Oaks: Sage Publications, 2003.
- [37] D. N. Card, “Learning from our mistakes with defect causal analysis,” *IEEE Software*, vol. 15, no. 1, pp. 56–63, Jan 1 98. [Online]. Available: <https://ieeexplore.ieee.org/document/646883>
- [38] G. Ooi and C. Soh, “Developing an activity-based costing approach for system development and implementation,” *SIGMIS Database*, vol. 34, no. 3, pp. 54–71, Aug. 2003. [Online]. Available: <http://doi.acm.org/10.1145/937742.937748>
- [39] “What is itil (information technology infrastructure library)?” 2019, [Accessed 01-November-2019]. [Online]. Available: <https://searchdatacenter.techtarget.com/definition/ITIL>

- [40] A. Steinacker, A. Ghavam, and R. Steinmetz, “Metadata standards for web-based resources,” *IEEE Multimedia*, vol. 8, no. 1, pp. 70–76, 2001. [Online]. Available: <http://www.dsc.ufcg.edu.br/~garcia/cursos/TEICOPIN/metadataWE.pdf>
- [41] D. A. Specification, “Deployment and Configuration of Component-based Distributed Applications Specification,” no. June, 2003. [Online]. Available: <https://staff.info.unamur.be/ven/CIS/OMG/Deployment%20and%20Configuration%20of%20Component-based%20Distributed%20Applications.pdf>
- [42] C. Ebert, R. Dumke, M. Bundschuh, and A. Schmietendorf, *Best Practices in Software Measurement*. SpringerVerlag, 2004.

Siglas

BA Bussiness Analysis. 4, 20

CMMI Capability Maturity Model Integration. 5, 6, 16, 21, 24, 25

CRM Customer Relationship Management. 4, 15–18

CROWNWeb Consolidated Renal Operations in Web-enabled Network. 2

EHR Electronic Health Record. 2, 16

ITIL Information Technology Infrastructure Library. 15

KEDB Known Error Database. 15

SaaS Software as a Service. 2, 4, 16

SDLC Software Development Life Cycle. 4, 15–18, 29, 34, 35

SEI Software Engineering Institute. 1

SLA Service Level Agreement. 2, 4