



**UNIVERSIDAD DE CHILE**  
**FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS**  
**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**

# **MEJORAS DE UN PROCESO DE DESARROLLO USANDO PRÁCTICAS ÁGILES**

TESIS PARA OPTAR AL GRADO DE  
MAGISTER EN TECNOLOGÍAS DE LA INFORMACIÓN.

GERMÁN MARIO ESPINOZA ZUÑIGA

PROFESOR GUÍA:

MARÍA CECILIA BASTARRICA PIÑEYRO

MIEMBROS DE LA COMISIÓN:

ALEXANDRE BERGEL

ALEJANDRO LAGOS SANTELICES

RENÉ NOËL LÓPEZ

SANTIAGO DE CHILE

2017

## Resumen

---

El presente trabajo se trata de la incorporación de prácticas ágiles en una compañía proveedora de soluciones de software. Esta compañía se llama **Telefónica MAKing Solutions (TMAS)**, la cual es encargada de ser un colaborador en los negocios de Telefónica en Latinoamérica.

Hay requerimientos que llegan desde los clientes, de menor magnitud para la compañía que resultan ineficientes si se abordan por el actual proceso de desarrollo.

El equipo de Soporte Mejoras, que es donde se desarrolla este trabajo, es un equipo del área de Soporte encargado de realizar Gestión de Problemas. Ha estado incorporando prácticas ágiles en su trabajo, lo que ha hecho que áreas encargadas de la Gestión de los Clientes, vean a este equipo como posible apoyo para implementar los requerimientos medianos y pequeños, ya que las estimaciones resultan más acordes al tamaño de la solicitud, y hace que los clientes acepten estas implementaciones.

Por lo tanto, el objetivo de este trabajo es incorporar prácticas ágiles de las metodologías XP y Scrum al proceso de desarrollo de la compañía y evaluar si estas logran un efecto positivo o negativo. Por lo que el alcance del trabajo es establecer un proceso de desarrollo utilizando prácticas ágiles acorde a la realidad de la compañía, el cual sirva como soporte de las actividades del equipo.

La principal medida del resultado del trabajo es la Tasa de Rechazo, que es el porcentaje de rechazo por errores de las entregas que se liberan al cliente, la medición se realizó semestral. La compañía definió que un porcentaje semestral mayor al 20% se considera que las soluciones entregadas al cliente son de mala calidad. El primer semestre donde se aplicó el trabajo se tuvo un resultado de 26,2% de rechazo y el último semestre en donde se aplicó la medición para este trabajo fue de un 8,3 % lo cual mejoró la calidad de los entregables.

Los primeros cálculos el resultado fue de mala calidad, según lo que definió la compañía que una Tasa de Rechazo mayor a 15% es considerado un servicio de mala calidad, lo cual fue mejorando con el tiempo.

## **Dedicatoria y Agradecimientos**

---

*Dedicado a mi Madre Dina Zúñiga Flores, a quien le debo todo.*

*Agradecimientos para la Profesora Cecilia Bastarrica, por su conducción en este trabajo.*

*Agradecimientos a Edmundo Martínez, y a la empresa TMAS, por la confianza y por permitirme realizar este trabajo en el equipo de Soporte Mejoras.*

*Agradecimientos a Jorge Retamal, Pedro Becker y David Veloz por su constante preocupación.*

*Agradecimientos muy especiales a Marcela Gutiérrez Pareja, por toda la ayuda, motivación y empuje que me ha dado siempre.*

# Tabla de Contenido

---

Resumen .....	i
Dedicatoria y Agradecimientos.....	ii
Tabla de Contenido.....	iii
Índice de Figuras .....	v
Índice de Tablas.....	vi
1 Introducción.....	1
1.1 Contexto.....	1
1.1.1 Misión de la Empresa .....	2
1.1.2 Estructura de la Compañía.....	2
1.2 El Problema .....	6
1.3 Objetivos.....	7
1.3.1 Objetivo General.....	7
1.3.2 Objetivos Específicos.....	7
1.4 Metodología.....	7
2 Marco Teórico.....	8
2.1 Ingeniería de Software.....	8
2.2 Metodologías Tradicionales.....	9
2.3 Metodologías Ágiles .....	10
2.3.1 XP.....	14
2.3.2 Scrum.....	18
2.4 Calidad del Software .....	20
2.5 Métricas.....	21
3 Solución Propuesta.....	23
3.1 Proceso Inicial.....	23
3.2 Proceso Modificado.....	28
3.2.1 Roles.....	29
3.2.2 Actividades .....	31
3.2.3 Revelamiento.....	32
3.2.4 Exploración.....	35
3.2.5 Planificación .....	46

3.2.6	Construcción .....	52
3.2.7	Producción .....	61
4	Ejecución del Proceso.....	72
4.1	Envíos hacia el Cliente.....	73
4.2	Tasa de Cumplimiento .....	76
4.3	Calidad .....	77
4.4	Evaluación Cualitativa de la Incorporación de Prácticas Ágiles .....	79
5	Conclusión .....	92
	Bibliografía .....	95
	Anexos .....	97
A.	PPM de HP.....	97
B.	Tablero Visual del Equipo de Mejoras.....	98
C.	Plantilla de Planificación.....	100
D.	Plantilla de Historias de Usuario .....	101
E.	Estimación de la Solución.....	103
F.	Plantilla de Estimación Final .....	105
G.	Plantilla de Casos de Prueba .....	107
H.	Plantilla de Documento ADP .....	108
I.	Proceso del Área de Desarrollo .....	109

# Índice de Figuras

---

Figura 1. Distribución Regional de las Operadoras de Movistar relacionadas con TMAS. ....	1
Figura 2. Organigrama General y Funcional de la Compañía. ....	3
Figura 3. Funciones del Área de Soporte. ....	4
Figura 4. Etapas de la Cascada. ....	9
Figura 5. Proceso Inicial del Equipo. ....	27
Figura 6. Diagrama General del Proceso Ágil. ....	28
Figura 7. Etapa de Revelamiento. ....	32
Figura 8. Etapa de Exploración. ....	35
Figura 9. Etapa de Planificación. ....	46
Figura 10. Etapa de Construcción de una Iteración. ....	53
Figura 11. Etapa de Producción. ....	61
Figura 12. Iteraciones entre Soporte Mejoras y los Dueños del Producto por PPM. ....	73
Figura 13. Kanban del Equipo. ....	99
Figura 14. Plantilla de Planificación. ....	100
Figura 15. Flujo de Trabajo de TMAS. ....	109
Figura 16. Etapa de Gestión Previa. ....	110
Figura 17. Etapa de Desarrollo. ....	111
Figura 18. Etapa de Implantación. ....	112

# Índice de Tablas

---

Tabla 1 Descripción Funcional de la Compañía. ....	4
Tabla 2 Funciones de los Equipos de Soporte.....	5
Tabla 3 Metodologías Ágiles vs Tradicionales. ....	13
Tabla 4 Roles del Proceso Inicial.....	24
Tabla 5 Etapas del Procesos Inicial.....	26
Tabla 6 Roles del Proceso Ágil Modificado.....	30
Tabla 7 Etapas del Proceso Ágil Modificado. ....	32
Tabla 8 Etapa de Revelamiento.....	32
Tabla 9 Recopilar Información.....	33
Tabla 10 Ingresar Solicitud al Tablero.....	34
Tabla 11 Analizar una solicitud.....	36
Tabla 12 Asignar Solicitud.....	37
Tabla 13 Analizar.....	38
Tabla 14 Conclusión del Análisis.....	39
Tabla 15 Escribir las Historias de Usuario.....	40
Tabla 16 Estimar la Solución.....	41
Tabla 17 Obtener la Cantidad Total Jornadas.....	43
Tabla 18 Inviabilidad de la Solución.....	43
Tabla 19 Cierre del Ticket de PPM de Mejoras.....	44
Tabla 20 Responder la Solicitud.....	45
Tabla 21 Etapa de Planificación.....	47
Tabla 22 Registrar la Respuesta del Dueño del Producto.....	48
Tabla 23 Planificación de las Iteraciones.....	49
Tabla 24 Reunión de Planificación en el Equipo.....	50
Tabla 25 Envío de Correo de la Planificación.....	51
Tabla 26 Etapa de Construcción de una Iteración.....	53
Tabla 27 Escribir Casos de Prueba.....	54
Tabla 28 Inspección de Casos de Prueba.....	55
Tabla 29 Implementación de la Solución.....	56
Tabla 30 Documentar la Solución.....	57
Tabla 31 Certificar la Solución.....	59
Tabla 32 Formalizar la Solución.....	60
Tabla 33 Revisión de la Iteración.....	60
Tabla 34 Etapa de Producción.....	62
Tabla 35 Instalación en Preproducción.....	64
Tabla 36 Pruebas de Aceptación del Cliente.....	65
Tabla 37 Resolución de Errores.....	66

Tabla 38 Paso a Producción. ....	67
Tabla 39 Monitorear la Solución. ....	68
Tabla 40 Actualización de Ambientes. ....	70
Tabla 41 Retrospectiva. ....	71
Tabla 42 Iteraciones con el Cliente en PPM. ....	74
Tabla 43 Período donde se Aplica el Envío. ....	77
Tabla 44 Cantidad de Iteraciones Realizadas. ....	78
Tabla 45 Resultados de la Tasa de Rechazo en el Período de Trabajo. ....	79
Tabla 46 Cualitativa de las Prácticas XP que se Utilizaron. ....	85
Tabla 47 Cualitativa de las Prácticas XP que se Utilizaron. ....	86
Tabla 48 Evaluación de las Prácticas de Scrum que se Utilizaron. ....	90
Tabla 49 Evaluación de las Prácticas de Scrum que no se Utilizaron. ....	91
Tabla 50 Plantilla de HU. ....	101



# 1

## 1 Introducción

### 1.1 Contexto

Telefónica **MA**king **S**olutions (TMAS) es una empresa de Tecnologías de Información dependiente del Grupo Telefónica en España, cuya visión como empresa es ser reconocida como colaboradora de excelencia a los negocios del Grupo Telefónica.

TMAS es la unidad regional de referencia en prácticas de desarrollo e implantación de sistemas de información, formando parte de la Dirección de TI de Telefónica Latinoamérica (Telefónica LATAM).

Las operadoras Movistar, que tienen una relación contractual con TMAS, son Argentina, Chile, Colombia, Ecuador, El Salvador, Guatemala, México, Nicaragua, Panamá, Uruguay y Venezuela.

TMAS está constituida por un centro de competencias ubicado en Santiago de Chile.



Figura 1. Distribución Regional de las Operadoras de Movistar relacionadas con TMAS.

### **1.1.1 Misión de la Empresa**

La misión de la empresa es asegurar la prestación eficiente de servicios de TI, identificando y aprovechando las oportunidades que contribuyan al cumplimiento de los objetivos de Telefónica en la región.

TMAS presta servicios a las operadoras Movistar de Latinoamérica, asociados a cada una de las etapas del ciclo de vida de operaciones de Negocios y Sistemas de Software, de la suite de aplicaciones llamada "Sistema Comercial Latinoamericano (desde ahora SCL)", tanto en la perspectiva lógica como en la física:

- ❖ Análisis de procesos de negocio y requisitos funcionales.
- ❖ Análisis y desarrollo de software.
- ❖ Gestión de cambios en el software y configuración de productos.
- ❖ Implantación de soluciones y gestión del cambio en el operador.
- ❖ Soporte a la operación.

Además de los anteriores, TMAS entrega en forma implícita otros servicios como:

- ❖ Distribución de mejores prácticas que se transportan de unas operadoras a otras.
- ❖ Distribución de costos en proyectos comunes.
- ❖ Reutilización de software.
- ❖ Configuración centralizada de productos y versiones.
- ❖ Recuperación de software operacional ante pérdidas catastróficas.
- ❖ Unificación de informes y métricas de gestión de Soluciones TI.

### **1.1.2 Estructura de la Compañía**

Como se representa en la figura 2, la compañía está compuesta por tres áreas las cuales se distribuyen las labores de la organización. En la tabla bajo la figura 2, se encuentra una descripción de cada área.

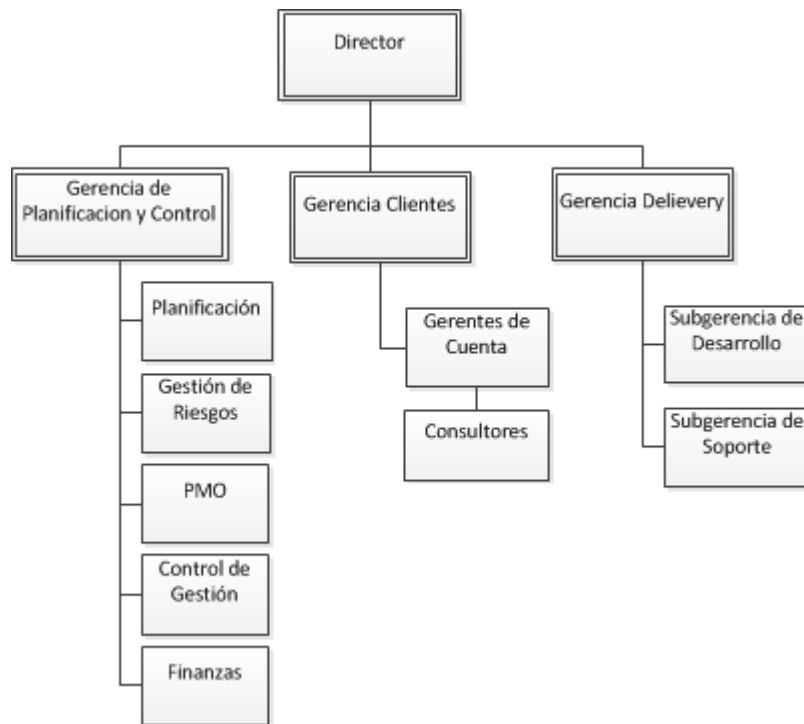


Figura 2. Organigrama General y Funcional de la Compañía.

Nombre de Área	Descripción	Nombre de subáreas	Descripción
Planificación y Control	Área encargada de la parte financiera de la empresa y del seguimiento del trabajo de la compañía.	Planificación	Administración del portafolio de proyectos, asignación de roles en el sistema.
		Gestión de Riesgos	Administración de riesgos en los proyectos.
		PMO	Visión general de proyectos.
		Control de Gestión	A cargo de los indicadores de proyectos y del servicio de soporte que están expresados en los contratos que se realizan con los clientes de la compañía.

		Finanzas	A cargo de los cobros por hito de la ejecución de los proyectos, y de los gastos de la compañía.
Clientes	Área encargada de la atención a clientes, recepción de necesidades de los clientes e implantación de proyectos.	Gerentes de cuenta	Encargados de gestionar al cliente, velar por sus necesidades y negociar los servicios.
		Consultores	Encargados de analizar los requerimientos e implantar proyectos.
Delievery	Área encargada de ejecutar los acuerdos que el área de clientes cierra.	Desarrollo	Ejecución y certificación de proyectos evolutivos.
		Soporte	Continuidad operacional de las aplicaciones de la suite de la compañía.

Tabla 1 Descripción Funcional de la Compañía.

Específicamente este trabajo se llevará a cabo en el Área de Soporte, por lo cual esta será descrita con mayor detalle. La siguiente figura muestra las funciones que tiene asignadas el área:

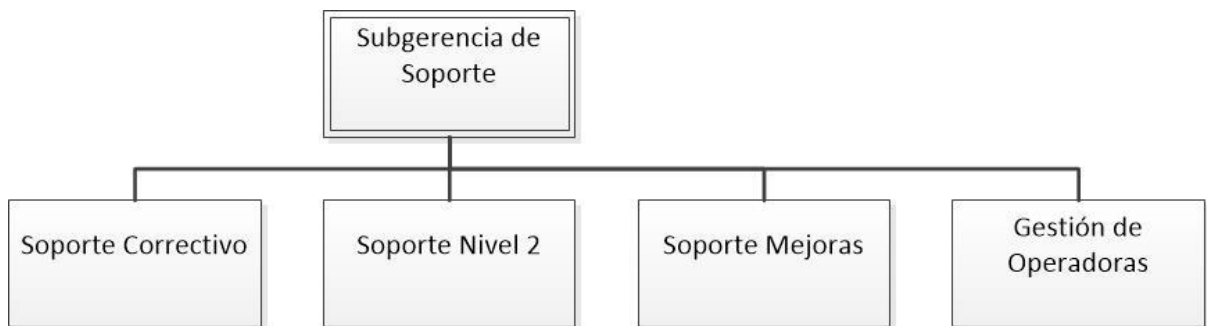


Figura 3. Funciones del Área de Soporte.

Cada uno de estos equipos tiene las siguientes funciones.

Equipo de Soporte	Funciones
Soporte Nivel 2	<p>Encargado de la continuidad operacional, de los procesos, configuraciones, consultas. Este equipo es el que recibe las incidencias del cliente y, si se trata de un error de software, lo deriva a Soporte Correctivo (que sería un Nivel 3).</p> <p>También es el encargado de gestionar las pruebas de las soluciones de software que genera Soporte Correctivo.</p>
Soporte Correctivo	<p>Encargado de la continuidad operacional de las aplicaciones en lo que se refiere a errores del software.</p> <p>Genera los documentos RCA<sup>1</sup>, según la recurrencia de las incidencias.</p>
Soporte Mejoras	<p>Encargado de realizar Gestión Problemas, dar soluciones de fondo a los problemas de la suite.</p> <p>Analiza los RCA.</p>
Gestión de Operadoras	<p>Encargados de velar por los KPI<sup>2</sup> fijados por contrato con los clientes, gestionar las excepciones de los indicadores, y velar por la correcta atención de soporte ante las operadoras.</p>

Tabla 2 Funciones de los Equipos de Soporte.

---

<sup>1</sup> RCA: acrónimo que corresponde a Root Cause Analysis, en español Análisis de Causa Raíz. Es un método para la resolución de problemas que intenta evitar la recurrencia de un problema o defecto a través de identificar sus causas. Cuando un incidente es reiterativo, se busca la causa de este problema.

<sup>2</sup> KPI: acrónimo que corresponde a Key Performance Indicator, en español indicador clave de rendimiento. Es una medida del nivel del desempeño de un proceso. El valor del indicador está directamente relacionado con un objetivo fijado de antemano y normalmente se expresa en porcentaje.

## 1.2 El Problema

Existen requerimientos que, dadas sus características, son de menor magnitud (estas características están relacionadas al esfuerzo que implica realizar la entrega de la solución). Se consideran de menor magnitud a los requerimientos que, desde su construcción hasta su liberación, no sean más de 45 jornadas de esfuerzo. Estos requerimientos resultan ineficientes si se abordan con el actual procedimiento estándar (Anexo A9), provocando que estos desarrollos no sean aceptados por los clientes de la compañía, producto de su costo. Esto se debe a la cantidad de documentación interna y las revisiones que generan información que no es vista por el cliente, ya que se aplica el mismo proceso a todos los requerimientos independiente de su alcance. Por lo tanto, se requiere poder dar una estimación, plazos y costos más acorde a la envergadura de los requerimientos, para que de esta manera nuestros clientes acepten realizar esta solución con TMAS.

El equipo de Soporte Mejoras del área de Soporte es el encargado de realizar Gestión de Problemas, y es donde se aplicará este trabajo. Actualmente este equipo está incorporando prácticas ágiles en el desarrollo de su trabajo. Por lo tanto, el problema que se intenta resolver es adoptar prácticas ágiles a según la realidad de la empresa y evaluar si este trabajo tiene una calidad acorde a lo que la compañía requiere. La adopción de metodologías ágiles ha mostrado en teoría ser una buena manera de gestionar desarrollos producto de su flexibilidad y velocidad<sup>3</sup>. Sin embargo, no se cuenta con un mecanismo que permita evaluar por medio de indicadores, si el desempeño de estas prácticas es satisfactorio en este contexto o si están cumpliendo su cometido.

Con esta información se podría saber si las prácticas ágiles son o no una alternativa efectiva y eficiente de desarrollo de este tipo de aplicaciones de software para esta Compañía.

---

<sup>3</sup> Libro Scrum Manager, Gestión de Proyectos. Pag. 36, "Escenario de Desarrollo en los 80".

## 1.3 Objetivos

### 1.3.1 Objetivo General

Definir un proceso de desarrollo de software para la compañía que incluya prácticas ágiles y que permita no superar el 15% de rechazos de los entregables.

### 1.3.2 Objetivos Específicos

- ❖ Definir y documentar un proceso de desarrollo de software eficiente para proyectos de menor envergadura que incluya prácticas ágiles, para el equipo de Soporte Mejoras de TMAS.
- ❖ Definir métricas para medir la calidad del trabajo realizado.
- ❖ Evaluar la conveniencia de aplicar prácticas ágiles en proyectos pequeños y medianos de desarrollo de software en la compañía. En base a los resultados de la calidad de los entregables y sobre una evaluación cualitativa de los directivos de la compañía.

## 1.4 Metodología

Los pasos que se realizarán en este trabajo serán:

- ❖ Documentar el proceso actual de desarrollo de software.
- ❖ En una reunión de equipo obtener el flujo de trabajo. Esta es la etapa inicial del ciclo, se valida cuál es el flujo de trabajo que se tiene en el equipo. Esta acción se realiza una vez al año.
- ❖ Se revisa el proceso actual con el flujo de trabajo que tiene el equipo para validar la diferencia entre estos para verificar posibles partes del proceso que sea necesario cambiar o reforzar.
- ❖ Se valida la incorporación de métricas o el cambio de las actuales.
- ❖ Se realizan los cambios en el proceso y se documenta.
- ❖ Se utiliza el nuevo proceso y se recolectan los datos y se mide. Se toma un año como para medir los cambios con las métricas establecidas.
- ❖ Se realiza una reunión de equipo y se validan los resultados obtenidos.

# 2

## 2 Marco Teórico

---

De nada sirven buenas prácticas y herramientas si no se proveen directivas para su aplicación. Las metodologías tradicionales tienen un marcado énfasis en el control del proceso mediante una rigurosa definición de roles, actividades y artefactos, incluyendo modelado y documentación detallada en las primeras etapas del proyecto. Este esquema más estructurado para abordar el desarrollo de software muestra ser efectivo en proyectos de gran tamaño (respecto a tiempo y recursos), donde por lo general se exige un alto grado de reuniones y ceremonias en el proceso. Sin embargo, este enfoque no resulta ser el más adecuado para muchos de los proyectos en donde el entorno del sistema es muy cambiante, y en donde se exige reducir drásticamente los tiempos de desarrollo, pero manteniendo una alta calidad. Ante las dificultades para utilizar metodologías tradicionales con estas restricciones de tiempo y flexibilidad, muchos equipos de desarrollo se resignan a prescindir del “buen hacer” de la ingeniería del software, asumiendo el riesgo que ello conlleva.

### 2.1 Ingeniería de Software<sup>4</sup>

El software es un elemento clave para los sistemas y productos informáticos. Entre algunas de las características del software están:

- ❖ Se desarrolla, no se fabrica: se utiliza un modelo de proceso de desarrollo que comprende análisis, diseño, desarrollo, implementación y evaluación para obtener un producto de calidad.
- ❖ El software durante su vida sufre cambios por lo que es probable que surjan fallos y defectos que si no se corrigen provocan que el software se vaya degradando
- ❖ A medida que el software evoluciona se crean estándares de diseño.

---

<sup>4</sup> Ingeniería de Software, Alejandra Aparicio, 2012. Pag. 9.



- ❖ El software debe diseñarse e implementarse para que pueda ser reutilizable.

La ingeniería de software es una disciplina que integra procesos, metodologías y herramientas para el desarrollo de software.

## 2.2 Metodologías Tradicionales

Una de las metodologías de desarrollo usadas en la compañía, es la metodología cascada, la cual consiste en realizar las etapas del proyecto de forma independiente y lineal (Ver figura 4).

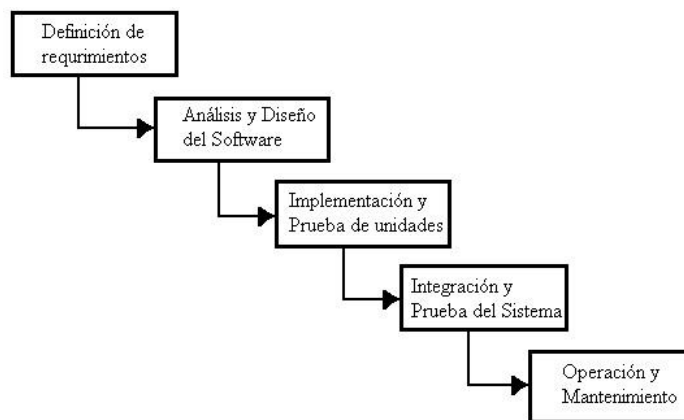


Figura 4. Etapas de la Cascada.

Dentro de las ventajas que tiene esta metodología, están:

- ❖ Es eficiente para dominios conocidos, y cuando se sabe que no habrá cambios en los requerimientos.
- ❖ Al ser lineal es simple de implementar.
- ❖ Se puede tener en parte un equipo de desarrollo menos capacitado ya que pocas personas toman las decisiones.

La metodología cascada tiene como principal desventaja su limitada capacidad de respuesta a los cambios y el tiempo que el negocio debe esperar para obtener un producto funcional. Además, es complicado poder documentar la totalidad de los requerimientos al inicio de los proyectos y que cumplan satisfactoriamente las necesidades de los usuarios.

Conforme el proyecto avanza, el costo para remediar errores de etapas anteriores suele ser muy alto, principalmente porque el equipo de trabajo debe

regresar una o más etapas para corregir el problema. Otra de las características importantes de este enfoque es el alto costo al implementar un cambio y el no ofrecer una buena solución para proyectos donde el entorno es volátil.

La necesidad de mejorar el proceso y llevar los proyectos al objetivo deseado hizo que se importase los fundamentos de metodologías existentes en otras áreas y adaptarlas al desarrollo de software.

Entre las principales metodologías tradicionales tenemos a RUP5, que centra su atención en llevar una documentación exhaustiva de todo el proyecto y en cumplir con un plan de proyecto, definido todo esto, en la fase inicial del desarrollo del proyecto.

Dentro de las ventajas de RUP, tenemos:

- ❖ Promueve la reusabilidad.
- ❖ Reduce la complejidad del mantenimiento (extensibilidad y facilidad de cambios).
- ❖ Riqueza semántica.
- ❖ Disminuye la brecha semántica entre la visión interna y la visión externa del sistema.
- ❖ Facilita la construcción de prototipos.

Las metodologías tradicionales se focalizan en documentación, planificación y procesos (plantillas, técnicas de administración, revisiones, etc.).

## 2.3 Metodologías Ágiles

Por estar orientadas para proyectos pequeños, las metodologías ágiles constituyen una solución a medida, aportando una elevada simplificación que a pesar de ello no renuncia a las prácticas esenciales para asegurar la calidad del producto.

Las metodologías ágiles es uno de los temas recientes en ingeniería de software que están acaparando gran interés. La comunidad de la ingeniería del software está viviendo con intensidad un debate abierto entre los partidarios de las metodologías tradicionales y aquellos que apoyan las ideas

---

5 Artículo Metodologías Tradicionales vs Metodologías Ágiles, de Roberth G. Figueroa , Camilo J. Solís y Armando A. Cabrera. De la Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación.

que provienen del "Manifiesto Ágil"<sup>6</sup>. La curiosidad que siente la mayor parte de los ingenieros de software, profesores, e incluso alumnos, sobre las metodologías ágiles hace prever una fuerte proyección industrial. Por un lado, para muchos equipos de desarrollo el uso de metodologías tradicionales resulta muy lejano a su forma de trabajo actual considerando las dificultades de su adopción e inversión asociada en formación y herramientas. Por otro, las características de los proyectos para los cuales las metodologías ágiles han sido especialmente pensadas se ajustan a un amplio rango de proyectos industriales de desarrollo de software: aquellos en los cuales los equipos de desarrollo son pequeños, con plazos reducidos, requisitos volátiles, y/o basados en nuevas tecnologías.

El punto de partida es el Manifiesto Ágil, que resume la filosofía ágil. Según el Manifiesto se valora:

- ❖ Al individuo y las interacciones del equipo de desarrollo por sobre el proceso y las herramientas. La gente es el principal factor de éxito de un proyecto software. Es más importante construir un buen equipo que construir el entorno. Muchas veces se comete el error de construir primero el entorno y esperar que el equipo se adapte naturalmente. Es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.
- ❖ Desarrollar software que funciona más que conseguir una buena documentación. La regla por seguir es no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante. Estos documentos deben ser cortos y centrarse en lo fundamental.
- ❖ La colaboración con el cliente es más importante que la negociación de un contrato. Se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.
- ❖ Responder a los cambios más que seguir estrictamente un plan. La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta sino flexible y abierta.

---

<sup>6</sup> Manifiesto Ágil: <http://agilemanifesto.org/iso/es/manifesto.html>

Los métodos ágiles de desarrollo utilizan un proceso rápido de desarrollo iterativo e incremental con altos niveles de comunicación y participación de los clientes. El enfoque ágil es más orientado a las personas en lugar de los procesos. Esto significa que depende en gran medida de las habilidades individuales.

Los dos métodos ágiles más populares son Extreme Programming (XP) y Scrum.

- ❖ XP<sup>7</sup> es un conjunto de principios y prácticas que tiene por objeto permitir el desarrollo de software con éxito a pesar de que los requerimientos estén mal documentados o que cambien constantemente, en equipos de trabajo pequeños y medianos.
- ❖ Scrum<sup>8</sup> tiene como objetivo gestionar el proceso de desarrollo a través de un enfoque empírico aplicando las ideas de control de procesos industriales en el desarrollo de software.

La cultura organizacional es una de las tareas de estudio más importantes de la organización y es muy importante para el éxito en la implementación de un modelo ágil en el desarrollo de software.

En la siguiente tabla se recogen las principales diferencias de las metodologías ágiles con respecto a las tradicionales. Estas diferencias afectan no sólo al proceso en sí, sino también al contexto del equipo y su organización.

---

<sup>7</sup> Scrum & eXtreme Programming para programadores, Eugenia Bahit, 2012. Pag. 70. Introducción a la Programación Extrema.

<sup>8</sup> Scrum & eXtreme Programming para programadores, Eugenia Bahit, 2012. Pag. 30. Conociendo Scrum.

Metodologías Ágiles <sup>9</sup>	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Especialmente preparadas para cambios durante el proyecto.	Cierta resistencia a los cambios.
Impuestas internamente (por el equipo).	Impuestas externamente.
Proceso menos controlado, con pocos principios.	Proceso mucho más controlado, con numerosas políticas/normas.
No existe contrato tradicional o al menos es bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.	Grupos grandes y posiblemente distribuidos.
Pocos artefactos.	Más artefactos.
Los roles no están fijos por personas.	Cada persona desempeña un rol específico.
La documentación por modelos de la arquitectura de software no es relevante	La arquitectura del software es esencial y se expresa mediante modelos.

Tabla 3 Metodologías Ágiles vs Tradicionales.

---

<sup>9</sup> Metodologías Tradicionales vs Metodologías Ágiles, Roberth G. Figueroa, Camilo J. Solís, Armando A. Cabrera, Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación. Pag. 7.

### 2.3.1 XP

**EX**treme **P**rogramming (XP) originalmente propone 12 prácticas, las cuales se describen a continuación.

#### ***PRÁCTICA #1: Cliente en el Lugar***

En XP se requiere que el cliente participe activamente del proyecto, contando con la disponibilidad suficiente y necesaria para interactuar con el equipo en todas las fases del proyecto. Siempre es recomendable contar con más de una persona (por parte del cliente), asignada a trabajar de forma permanente con el equipo.

La comunicación cara a cara con el cliente es fundamental, ya que, a partir de ésta, el equipo avanzará más rápidamente en el proyecto, puesto que:

- ❖ Responderá todas sus dudas sobre el proyecto con el cliente y en el momento que éstas surjan.
- ❖ Se establecerán las prioridades a desarrollar en tiempo real, así como la resolución de conflictos.

Las personas asignadas por el cliente deben ser:

- ❖ Conocedores expertos de lo que se pretende producir.
- ❖ En lo posible, se pretende que sean usuarios reales del software.
- ❖ Personas que cuenten con la información suficiente sobre la aplicación y los objetivos del negocio, a fin de poder actuar con autoridad en la toma de decisiones.

#### ***PRÁCTICA #2: Semana de 40 horas***

El equipo no debe asumir actividades que le demanden un mayor esfuerzo del que humanamente se puede disponer. Cada persona debe trabajar como máximo 40 horas semanales. Aunque las 40 horas semanales no aseguran calidad, en este tiempo acotado se asume que el equipo desarrollará tanto software como sea posible y de buena calidad.

XP supone que un equipo descansado, sin esfuerzos desmedidos, logra un mejor resultado.

### ***PRÁCTICA #3: Metáfora***

A fin de evitar los problemas de comunicación que suelen surgir en la práctica entre técnicos y usuarios, XP propone el uso de metáforas, intentando hallar un punto de referencia que permita representar un concepto técnico con una situación en común con la vida cotidiana y real.

Para que una metáfora cumpla este objetivo, es fundamental:

- ❖ Hacer un paralelismo entre una funcionalidad del sistema y la vida real.
- ❖ Si ese paralelismo (metáfora) no se entiende inmediatamente, se deberá buscar otra metáfora (una metáfora sólo sirve cuando es comprendida de forma inmediata).

Muchas veces, puede ser necesario emplear más de una metáfora para explicar una misma característica del sistema. Incluso, en algunas oportunidades, puede no encontrarse metáforas que describan con precisión todas las características que se intenta explicar. Son momentos, en los que se requiere de un gran ingenio y creatividad.

### ***PRÁCTICA #4: Diseño Simple***

Esta práctica, deriva de un famoso principio técnico del desarrollo de software: KISS. Cuyas siglas derivan del inglés "Keep it simple, stupid!" -¡Mantenlo simple, estúpido!

Básicamente, el principio KISS, se trata de mantener un diseño sencillo, estandarizado, de fácil comprensión y refactorización. Puede resumirse en "hacer lo mínimo indispensable, tan legible como sea posible".

### ***PRÁCTICA #5: Refactorización***

La refactorización es una técnica que consiste en modificar el código fuente de un software sin afectar a su comportamiento externo.

Se propone primero "desarrollar lo mínimamente indispensable". Después, en el desarrollo de cada nueva funcionalidad, puede (y debe) ser necesario, realizar una refactorización del código a fin de lograr una mayor cohesión de éste, impidiendo redundancias.

La refactorización es la principal técnica propuesta por XP para evolucionar el código de la aplicación.

### ***PRÁCTICA #6: Programación de a Pares***

Se estila (aunque no de forma constante) programar en parejas de dos desarrolladores, los cuales podrán ir intercambiando su rol, en las sucesivas programaciones de a pares.

Dos programadores, sentados frente a un computador, cada uno cumpliendo un rol diferente que suelen ser:

- ❖ Uno escribe el código, mientras que otro lo va revisando;
- ❖ Un programador más avanzado, programa, mientras va explicando lo desarrollado a otro menos experto;
- ❖ Los dos programadores piensan en cómo resolver el código, y uno de ellos lo escribe.

De esa forma y según las necesidades de cada equipo, las funciones de cada programador podrán variar.

### ***PRÁCTICA #7: Entregas Cortas***

Se busca hacer entregas en breves lapsos de tiempo, incrementando pequeñas funcionalidades en cada iteración. Esto conlleva a que el cliente pueda tener una mejor experiencia con el software, ya que lo que deberá probar como nuevo, será poco, y fácilmente asimilable, pudiendo sugerir mejoras con mayor facilidad de implementación. Las entregas cortas, reducen la curva de aprendizaje sobre la aplicación.

### ***PRÁCTICA #8: Pruebas***

Para XP hay tres tipos de pruebas (test), propuestos:

- ❖ Test unitarios.
- ❖ Test de aceptación.
- ❖ Test de integración.



Los primeros, consisten en testear el código de manera unitaria (individual) mientras se va programando. Esta técnica es conocida como Unit Testing, la cual forma parte de la técnica TDD (Test-Driven Development – Desarrollo Guiado por Pruebas).

Los test de aceptación están más abocados a las pruebas de funcionalidad, es decir al comportamiento funcional del código. Estas pruebas, a diferencia de los Test Unitarios, son definidas por el cliente.

Los test de integración tienen por objeto integrar todos los módulos o unidades que conforman la aplicación a fin de validar el correcto funcionamiento de la misma.

### ***PRÁCTICA #9: Código Estándar***

Los estándares de escritura del código fuente son esenciales a la hora de programar. Permiten hacer más legible el código y más limpio a la vez de proveer a otros programadores una rápida visualización y entendimiento del mismo.

### ***PRÁCTICA #10: Propiedad Colectiva***

No existe un programador “dueño” de un determinado código o funcionalidad. La propiedad colectiva del código tiene por objetivo que todos los miembros del equipo conozcan “qué” y “cómo” se está desarrollando el sistema, evitando así, lo que sucede en muchas empresas, que existen “programadores dueños de un código” y cuando surge un problema, nadie más que él puede resolverlo.

### ***PRÁCTICA #11: Integración Continua***

La integración continua de XP propone que todo el código (desarrollado por los miembros del equipo) encuentre un punto de alojamiento común en el cuál deban enviarse los nuevos desarrollos, diariamente, después de correr los test de integración, a fin de verificar que lo nuevo, no “rompa” lo anterior.

Estos puntos de alojamiento en común suelen ser repositorios, los cuales pueden manejarse ampliando las ventajas de la integración, mediante software de control de versiones como Bazaar, Git o SVN, entre otros.

En este punto, los test de integración juegan un papel fundamental puesto que de ellos depende la integración de lo nuevo con lo ya desarrollado. Independientemente de los repositorios, jamás se logrará una integración continua real si no existen test de integración o si éstos fallan.

### **PRÁCTICA #12: Juego de Planificación**

La dinámica de planificación se lleva a cabo al inicio de la iteración, la cual suele ser la siguiente:

- ❖ El cliente presenta la lista de las funcionalidades deseadas para el sistema, escrita con formato de "Historia de Usuario", en la cual se encuentra definido el comportamiento de la misma con sus respectivos criterios de aceptación.
- ❖ El equipo de desarrollo estima el esfuerzo que demanda desarrollarlas, así como el esfuerzo disponible para el desarrollo (las iteraciones en XP suelen durar 1 a 4 semanas).
- ❖ El cliente decide qué Historias de Usuario desarrollar y en qué orden.

### **2.3.2 Scrum**

Scrum es una metodología para la gestión de proyectos relacionados con el desarrollo de software.

Scrum es un marco de trabajo iterativo e incremental para el desarrollo de proyectos, productos y aplicaciones. Estructura el desarrollo en ciclos de trabajo llamados Sprints. Son iteraciones de 1 a 4 semanas, y se van sucediendo una detrás de otra. Los Sprints son de duración fija – terminan en una fecha específica, aunque no se haya terminado el trabajo, y nunca se alargan. Se limitan en tiempo. Al comienzo de cada Sprint, un equipo multi-funcional selecciona los elementos (requisitos del cliente) de una lista priorizada. Se comprometen a terminar los elementos al final del Sprint. Durante el Sprint no se pueden cambiar los elementos elegidos (Theo Scrum Primer, 2009, pág. 5).

Un Sprint tiene tres características fundamentales:

- ❖ Una duración fija entre 1 a 4 semanas.

- ❖ Cada Sprint se ejecuta de forma consecutiva (sin tiempo muerto entre un Sprint y otro).
- ❖ El objetivo principal es transformar un conjunto de ítems requeridos por el cliente en un incremento de funcionalidad 100% operativa para el software.

“Todos los días el equipo se reúne brevemente para informar del progreso, y actualizan unas gráficas sencillas que les orientan sobre el trabajo restante. Al final del Sprint, el equipo revisa el Sprint con los interesados en el proyecto, y les muestra lo que han construido. La gente obtiene comentarios y observaciones que se pueden incorporar al siguiente Sprint. Scrum pone el énfasis en productos que funcionen al final del Sprint que realmente estén “hechos”; en el caso del software significa que el código esté integrado, completamente probado y potencialmente para entregar”. (The Scrum Primer, 2009, pág. 5).

Los equipos de Scrum definen tres roles bien diferenciados:

- ❖ El Scrum Master, responsable de asegurar los procesos.
- ❖ El Dueño de Producto, responsable de maximizar el valor del producto.
- ❖ El Equipo, responsable de realizar el trabajo.

La definición de bloques de tiempo iterativos (de 1 a 4 semanas) está destinada a crear continuidad y regularidad en las que se basarán las cinco ceremonias (reuniones) que intentan asegurar el cumplimiento de los objetivos:

1. Reunión de Planificación de la Entrega: El Dueño del Producto, indica la prioridad de las solicitudes.
2. Reunión de Planificación del Sprint: El objetivo de esta reunión es que el Dueño de Producto pueda presentar al equipo las historias de usuario prioritarias comprendidas en el Backlog de producto; que el equipo comprenda el alcance de las mismas mediante preguntas; y que ambos negocien cuáles pueden ser desarrolladas en el Sprint que se está planificando.
3. Reunión Diaria: Las reuniones diarias para Scrum, son "conversaciones" de no más de 5-15 minutos, que el Scrum Master tendrá al comienzo de cada día con todos los miembros del equipo. En esta conversación, el Scrum Master deberá ponerse al día de lo que cada miembro ha desarrollado (en la jornada previa), lo que hará en la fecha actual, pero por sobre todo, conocer cuáles impedimentos estén surgiendo, a fin de

resolverlos y que el equipo pueda continuar sus labores sin preocupaciones.

4. Reunión de Revisión: el equipo presentará al Dueño de Producto las funcionalidades desarrolladas. Las explicará y hará una demostración de ellas, a fin de que tanto Dueño de Producto como la eventual audiencia, puedan experimentarlas.
5. Reunión de Retrospectiva: El objetivo es "mirar hacia atrás (en retrospectiva)", realizar un análisis de lo que se ha hecho y sus resultados correspondientes, y decidir qué medidas concretas emplear, a fin de mejorar esos resultados.

Como herramientas aplicables a todos los procesos anteriores, Scrum emplea cuatro artefactos:

1. Backlog de Producto: una lista priorizada de todo lo que requiere el software.
2. Backlog de Sprint: una lista de tareas necesarias para convertir parte del Backlog de Producto, en incremento de funcionalidad del software.
3. Scrum taskboard: un tablero físico que permite la transparencia de los ítems de trabajos pendientes, en curso y terminados de un mismo Sprint.
4. Diagrama de Burndown: un gráfico que permite medir visualmente el progreso de los ítems del Backlog de Producto.

## 2.4 Calidad del Software

La calidad del software es una compleja combinación de factores, que variarán entre diferentes aplicaciones.

La norma UNE-EN ISO 8402<sup>10</sup>, expresa que "la calidad es el conjunto de propiedades y características de un producto o servicio que le confieren su aptitud para satisfacer unas necesidades explícitas o implícitas". Llevada esta definición al campo de la ingeniería de software, la IEEE Std 610, señala que "la calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o

---

<sup>10</sup> Modelos y Métricas para Evaluar Calidad de Software. Marcelo Estayno; Gladys Dapozo; Liliana Cuenca Pletch; Cristina Greiner.

expectativas del cliente o usuario". Por su parte, Pressman se refiere a la calidad del software como "la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, estándares de desarrollo explícitamente documentados y características implícitas que se espera de todo software profesionalmente desarrollado". Sommerville, sostiene que la calidad del software es un concepto complejo que no es directamente comparable con la calidad de la elaboración de productos.

Según la ISO 25000, la calidad del producto, junto con la calidad del proceso, es uno de los aspectos más importantes en el desarrollo de Software. La ISO/IEC 25000, proporciona una guía para el uso de la nueva serie de estándares internacionales llamada Requisitos y Evaluación de Calidad de Productos de Software (SQuaRE - System and Software Quality Requirements and Evaluation).

## 2.5 Métricas

La medición es un elemento clave en cualquier proceso de ingeniería. Las medidas se emplean para comprender mejor los atributos de los modelos que se crean y evaluar la calidad de los productos de la ingeniería. Una métrica contiene la definición de un método de medición o un método de cálculo y la escala asociada.

La definición de las métricas debe basarse en objetivos de medida claros. Aunque en la teoría relativa a las métricas de software se comenta la necesidad de la validación, no existen propuestas que integren todos los aspectos que es necesario tener en cuenta en dicha validación. El método consta de diversas fases que van desde la identificación de los objetivos y las hipótesis de trabajo hasta la aplicación y posterior retirada de una métrica. Este proceso consta de cinco etapas principales:

- ❖ **Identificación:** Se definen los objetivos que se persiguen a la hora de crear la métrica y se plantean las hipótesis de cómo se llevará a cabo la medición. Sobre los elementos de esta etapa (objetivos e hipótesis) se basarán todas las etapas siguientes. Como resultado de esta etapa se generan los requisitos que debe cumplir la métrica.
- ❖ **Creación:** Se realiza la definición de la métrica y su validación teórica y empírica.
- ❖ **Aceptación:** Una vez obtenida una métrica válida, suele ser necesario pasar por una etapa de aceptación de la métrica en la que se harán

pruebas en entornos reales, de manera que podamos comprobar si la métrica cumple los objetivos deseados dentro del campo de aplicación real.

- ❖ Aplicación: Una vez que tengamos una métrica aceptada, la utilizaremos dentro del campo de la aplicación para la que fue diseñada.
- ❖ Acreditación: Es la última etapa del proceso, que discurre en paralelo con la fase de aplicación y tiene como objetivo el mantenimiento de la métrica, de manera que se la pueda adaptar al entorno cambiante de aplicación.

Como consecuencia de esta etapa, puede que una métrica sea retirada porque ya no sea útil en el entorno en el que se aplica o que sea reutilizada para iniciar el proceso de nuevo.

Por ejemplo:

- ❖ Identificación: Medir los rechazos de los entregables implementados.
- ❖ Creación: Se determina y se define cómo se van a contar los rechazos realizados por el cliente.
- ❖ Aceptación: Se acuerda con las personas involucradas la manera de realizar la medida.
- ❖ Aplicación: Una vez aceptada la métrica, se comienza a medir.



## 3 Solución Propuesta

---

Existe actualmente un proceso implícito que se aplica en el desarrollo de software. Este trabajo de tesis pretende formalizar el proceso actual, modificarlo para incluir prácticas ágiles, y medir la efectividad de estos cambios.

### 3.1 Proceso Inicial

Inicialmente el proceso tiene cuatro roles los cuales se describen a continuación:

Roles	Descripción
Gestor de Mejoras	El equipo de trabajo que tiene como objetivo: <ul style="list-style-type: none"><li>❖ Analizar requerimientos.</li><li>❖ Estimar.</li><li>❖ Codificar.</li><li>❖ Certificar.</li><li>❖ Gestionar a clientes.</li><li>❖ Gestionar los errores en las pruebas.</li><li>❖ Inspección de documentos y código.</li><li>❖ Documentar requerimientos, casos de pruebas, etc.</li><li>❖ Gestionar Riesgos.</li></ul>
Dueño del Producto	Es quien solicita los requerimientos al equipo.

Responsable del Equipo	Es quien es responsable del trabajo del equipo, y realiza mejora continua al proceso.
Usuario	Representantes de los usuarios finales de las aplicaciones.

Tabla 4 Roles del Proceso Inicial.

Por otro lado, el proceso con el cual se comenzó el trabajo en el equipo consistía en una etapa de análisis que entregaba la solución a realizar, y después se realizaba la estimación a juicio experto, la cual es aprobada por el Dueño del Producto. Si la aprueba la solución se implementará y se entregará al Dueño de Producto para que se planifique las pruebas con los Usuarios y su posterior paso a producción con los Usuarios. Las actividades de este proceso inicial de muestran en la figura 5.

La descripción general del proceso es el siguiente:

Etapa	Descripción
Análisis	<p>El Dueño del Producto presenta una nueva solicitud al Responsable del Equipo. Éste asigna un Gestor de Mejoras para poder realizar el análisis de la solicitud.</p> <p>El Gestor de Mejoras resuelve las dudas que pudiese tener sobre la solicitud, canalizándolas con el Dueño del Producto.</p> <p>Una vez que ya no hay más dudas por resolver, se reúne el equipo para definir entre todos la posible solución a implementar.</p> <p>Si no hay una solución a implementar por software se informa esto al Dueño del Producto y se termina la solicitud. Pero si existe una solución a implementar esta se debe estimar.</p>
Estimación	En una reunión de equipo cada asistente a la reunión dice a viva voz cuánto cree que le tomaría implementar la solución, en jornadas. Se saca un promedio de las



	<p>jornadas, y el Responsable del Equipo, agrega el resto de las jornadas, como:</p> <ul style="list-style-type: none"> <li>❖ Ambientación.</li> <li>❖ Documentación.</li> <li>❖ Certificación.</li> <li>❖ Formalización.</li> </ul> <p>Con esta información el Responsable del Equipo genera la propuesta de solución y se hace una planificación con las fechas de la posible implementación de la solución.</p> <p>Si el Dueño del Producto no acepta el presupuesto de la solución, la solicitud se termina.</p>
<p>Construcción</p>	<p>Si se acepta el presupuesto, se debe proceder a revisar la planificación y definir la fecha definitiva de inicio y término entre el Responsable del Equipo y el Dueño del Producto. Se asignan además las tareas de la implementación y se informan a los Gestores de Mejoras involucrados.</p> <p>Se implementa la solución, se escriben casos de prueba y se certifica la solución.</p> <p>Se escribe la documentación a enviar al Dueño del Producto y se formaliza la solución, esto quiere decir, se versionan los cambios y se hacen los paquetes entregables.</p> <p>Se libera el paquete al Dueño del Producto.</p>
<p>Producción</p>	<p>El Dueño del Producto prepara la instalación del paquete en preproducción, y si falla el paquete es rechazado y se tiene que volver a hacer.</p> <p>Si se instala apropiadamente, se procede con las pruebas de aceptación con los Usuarios. Si hay fallas, el paquete es devuelto al equipo de desarrollo para su resolución generando un nuevo paquete que debe ser instalado nuevamente en preproducción.</p> <p>Cuando todas las pruebas pasan, el Dueño del Producto planifica el paso a producción.</p>

	<p>Una vez que el paquete pasa a producción y a solicitud del Dueño del Producto se puede monitorear el funcionamiento del cambio en producción.</p> <p>Cuando definitivamente el cambio queda en producción se actualizan los ambientes internos de la compañía, para que el resto de la empresa se dé por enterado del cambio y que estas piezas serían la última versión de las aplicaciones.</p>
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 5 Etapas del Procesos Inicial.

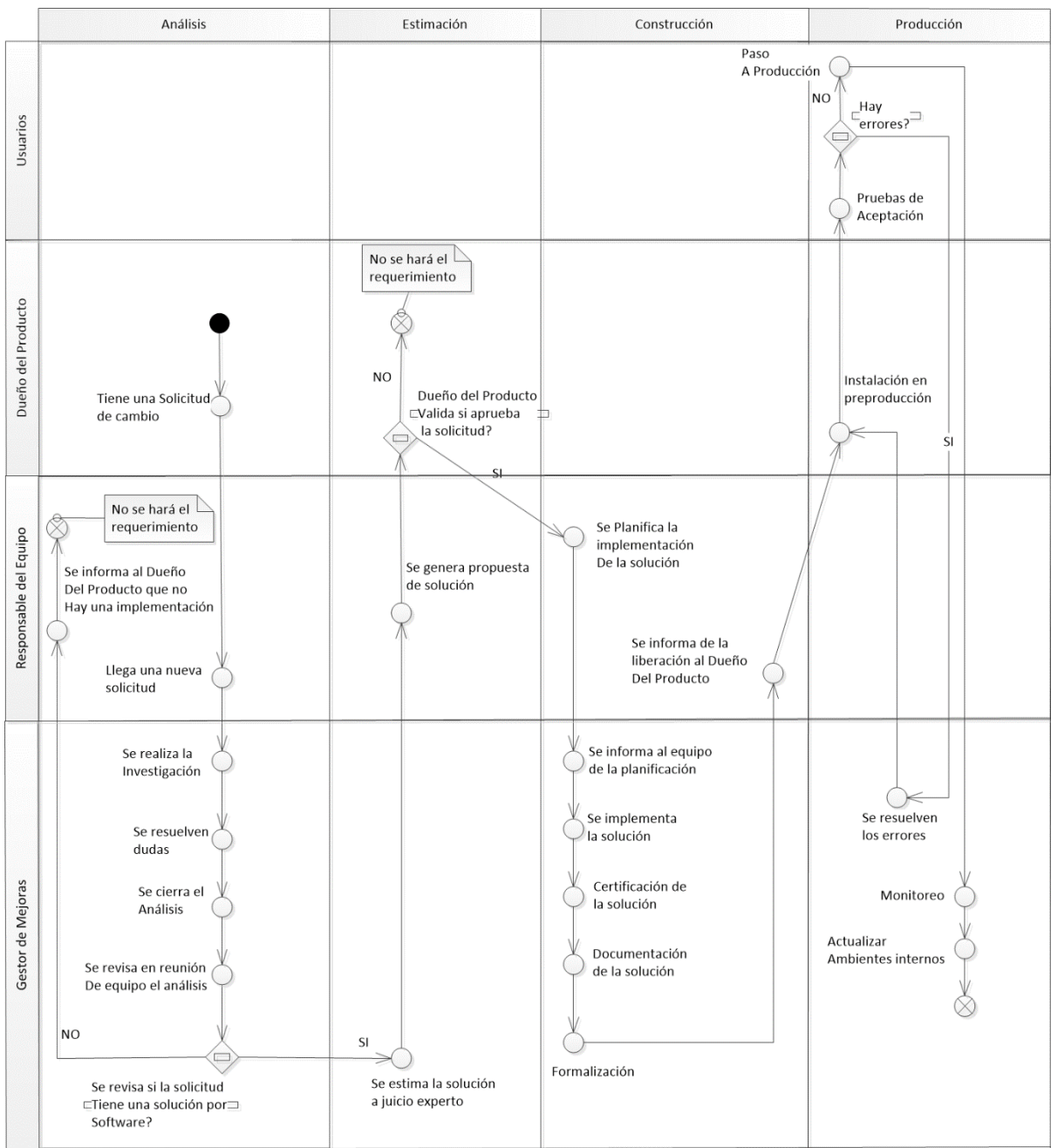


Figura 5. Proceso Inicial del Equipo.

### 3.2 Proceso Modificado

Como uno de los objetivos de este trabajo es la definición de un proceso con prácticas ágiles, en esta sección se mostrará el proceso resultante de las modificaciones del proceso inicial habiendo pasado por dos iteraciones de la metodología definida.

El proceso consta de 5 partes las cuales se detallan a continuación. En la siguiente figura se muestra el Diagrama General del proceso:

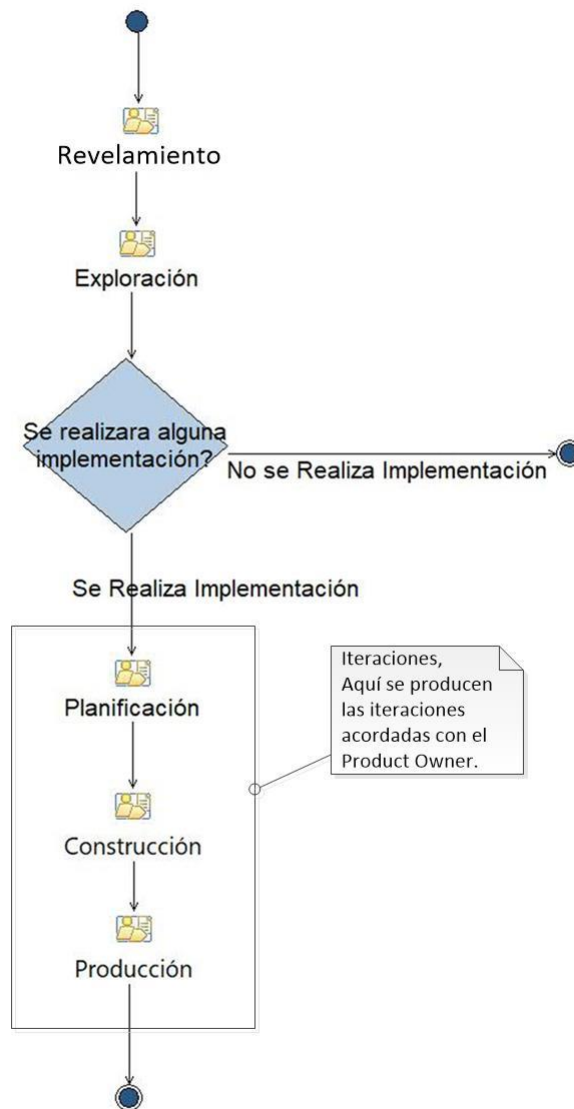


Figura 6. Diagrama General del Proceso Ágil.

### 3.2.1 Roles

Para poder describir las actividades de este proceso primero se describirán los Roles que están involucrados en este perfil (guiándose de los roles definidos en Scrum).

Rol	Descripción
<p>Gestor de Mejoras. Es el equipo de desarrollo definido en Scrum.</p>	<p>Es parte del equipo de desarrollo, y está encargado de:</p> <ul style="list-style-type: none"> <li>❖ Analizar.</li> <li>❖ Definir Soluciones.</li> <li>❖ Construir.</li> <li>❖ Certificar.</li> <li>❖ Inspeccionar.</li> <li>❖ Documentar.</li> <li>❖ Formalizar.</li> <li>❖ Implantar.</li> <li>❖ Monitorear.</li> </ul> <p>Las soluciones entregadas a los clientes.</p>
<p>Responsable de Equipo. Es el Scrum Master, definido en Scrum</p>	<p>Es el responsable del equipo de desarrollo (aquí se evita que este cargo se llame Jefe, para no tener una jerarquía innecesaria). Dentro de las actividades que tiene que realizar:</p> <ul style="list-style-type: none"> <li>❖ Liderar el Equipo de Desarrollo.</li> <li>❖ Verificar que el proceso se lleve a cabo.</li> <li>❖ Aplicar Mejoras Continuas.</li> <li>❖ Innovar.</li> <li>❖ Facilitador del Equipo.</li> <li>❖ Responsable de la reunión diaria.</li> <li>❖ Responsable de la reunión de Estimación.</li> <li>❖ Responsable de la planificación.</li> <li>❖ Responsable de la retrospectiva.</li> <li>❖ Responsable de cualquier actividad que realiza el equipo.</li> </ul>

Usuario del Sistema	Usuario del sistema SCL (siglas definidas en la página 2), el cual es conocedor del funcionamiento del Sistema, por lo cual puede certificar las soluciones entregadas por el Equipo de Desarrollo. Los Usuarios son definidos por el Cliente.
Dueño del Producto o Product Owner	El Dueño del Producto puede ser alguien de la compañía o también alguien del cliente si es que el Equipo de Desarrollo tiene contacto directo con el cliente. Tiene como actividades: <ul style="list-style-type: none"> <li>❖ Entrega los Requerimientos.</li> <li>❖ Priorización de los Requerimientos.</li> <li>❖ Responder las consultas del Equipo de Desarrollo.</li> <li>❖ Validar la solución que se dará por el Equipo de Desarrollo.</li> <li>❖ Acompañar a los Usuarios en las Pruebas de Aceptación.</li> </ul>
Experto del Módulo	Personas al interior de la compañía que son conocedoras de un módulo, tanto a nivel funcional y/o técnico. Estas personas son necesarias tenerlas presentes por consultas más especializadas, que ayudaran al equipo de desarrollo a realizar mejor su trabajo. Las metodologías ágiles promueven el tener al cliente en el lugar donde se está implementando la solución para resolver las dudas directamente, ahora en nuestro caso el cliente gestiona el tiempo de los Usuarios, por lo que en general nunca los tenemos para hacerles preguntas en el momento sino que son principalmente por correo o llamadas por teléfono, es aquí donde el Experto en el Módulo, provee información directa sobre el funcionamiento de la aplicación, ahora lo que no supe es información sobre la solicitud.

Tabla 6 Roles del Proceso Ágil Modificado.

### 3.2.2 Actividades

Las actividades que se realizan en este proceso son las siguientes:

Actividades	Descripción
Revelamiento	Tener conocimiento de una nueva solicitud. El Dueño del Producto, envía una solicitud al equipo para su análisis.
Exploración	<p>El equipo de Gestores de Mejoras analiza la solicitud para buscar una posible solución.</p> <p>De no tener una solución que pase por la implantación de software se responden estas razones al Dueño del Producto. Ahora si se puede implementar una solución, se propone al Dueño del Producto, las iteraciones que se pueden realizar y se queda a la espera de su aprobación u observación si es que se implementa la solución. Pero es en la etapa de Planificación donde se definen la cantidad final de iteraciones y las fechas de cada entrega.</p>
Planificación	<p>En una etapa inicial cuando el Dueño del Producto aprueba la implementación de la solución, se cierran las iteraciones las fechas de entrega de cada una, esto negociado con el Dueño del Producto.</p> <p>Cuando se inicia cada iteración se planifican las tareas y sus asignaciones de los Gestores de Mejoras que participaran en la iteración.</p>
Construcción	Se implementa la solución planificada para la iteración. Esta se certifica y se libera al Dueños del Producto.
Producción	El Dueño del Producto junto con los Usuarios, realizan las pruebas de aceptación. Si hay errores estos deben ser resueltos por el equipo de desarrollo, si estas son ejecutadas correctamente se pasa a producción el entregable de la iteración.

	Se monitorea el funcionamiento de la solución en producción y se cierra el ciclo de la iteración cuando toda la solución está estable.
--	----------------------------------------------------------------------------------------------------------------------------------------

Tabla 7 Etapas del Proceso Ágil Modificado.

Las tareas y roles involucrados en cada una de estas actividades es lo que se describirá en las siguientes secciones.

### 3.2.3 Revelamiento

La actividad de Revelamiento tiene el siguiente proceso:

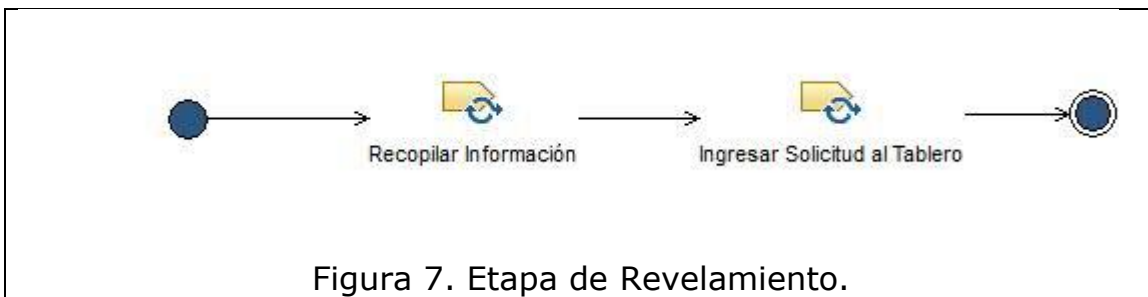


Figura 7. Etapa de Revelamiento.

Propósito	El Dueño del Producto, entrega al Equipo un tema para su análisis.
Descripción	<p>El Dueño del producto envía, en el formato que él estime conveniente, información sobre la solicitud que se requiere implementar una solución. No se impone una manera de enviar esta información.</p> <p>Con estos datos se crea un ticket en la aplicación PPM (Ver Anexo A.1); con esto se asigna un <b>Identificador</b> a la solicitud, y se sube la información entregada por el Dueño del Producto a PPM.</p>

Tabla 8 Etapa de Revelamiento.



Descripción de las tareas:

Descripción de la Tarea	Recopilar Información
Roles	<ul style="list-style-type: none"> <li>❖ Dueño del Producto.</li> <li>❖ Responsable del Equipo.</li> </ul>
Entradas	Nueva Solicitud: El dueño del producto envía por correo una nueva solicitud para ser tratada por el equipo de Soporte Mejoras. Esta solicitud debe ser enviada al Responsable del Equipo.
Salidas	Ticket en PPM, de tipo "Mejoras" (desde ahora PPM Mejoras).
Descripción	Llega una nueva Solicitud de parte de alguno de los Dueños del Producto, y se reúne todo tipo de información enviada.
Pasos	<ul style="list-style-type: none"> <li>❖ Reunir información: El Dueño del Producto envía información relacionada al nuevo requerimiento. Esta es enviada por correo con archivos adjuntos o solo texto en el correo. No importa el formato de la información enviada.</li> <li>❖ Generar ticket en PPM: El Responsable del Equipo genera un Requerimiento de Soporte de Tipo Mejoras en PPM. Con esto se ingresa la solicitud al sistema de gestión de la compañía, y se genera una identificación de la solicitud.</li> <li>❖ Subir información: Subir al ticket de Mejoras en PPM, la información reunida. Correos o archivos relacionados.</li> </ul>

Tabla 9 Recopilar Información.

Descripción de la Tarea	Ingresar Solicitud al Tablero
Roles	<ul style="list-style-type: none"> <li>❖ Responsable del Equipo.</li> <li>❖ Gestor de Mejoras.</li> </ul>
Entradas	PPM de Mejoras.
Salidas	Backlog (Anexo A.2).
Descripción	Son todas las solicitudes que llegan al equipo y que no han sido revisadas aún, por lo que pasan a estar pendientes.
Pasos	<ul style="list-style-type: none"> <li>❖ Escribir la nota en un post-it: En el post-it escribir, los siguientes datos: <ul style="list-style-type: none"> <li>• El código del cliente.</li> <li>• El número de ID de PPM.</li> <li>• Las siglas del módulo.</li> <li>• Descripción breve de la solicitud.</li> </ul> </li> <li>❖ Poner el post-it en el Tablero: Poner el post-it en el tablero en la sección Backlog.</li> </ul>
Práctica	Tablero visual o Kanban.

Tabla 10 Ingresar Solicitud al Tablero.

### 3.2.4 Exploración

La actividad de Exploración tiene el siguiente proceso:

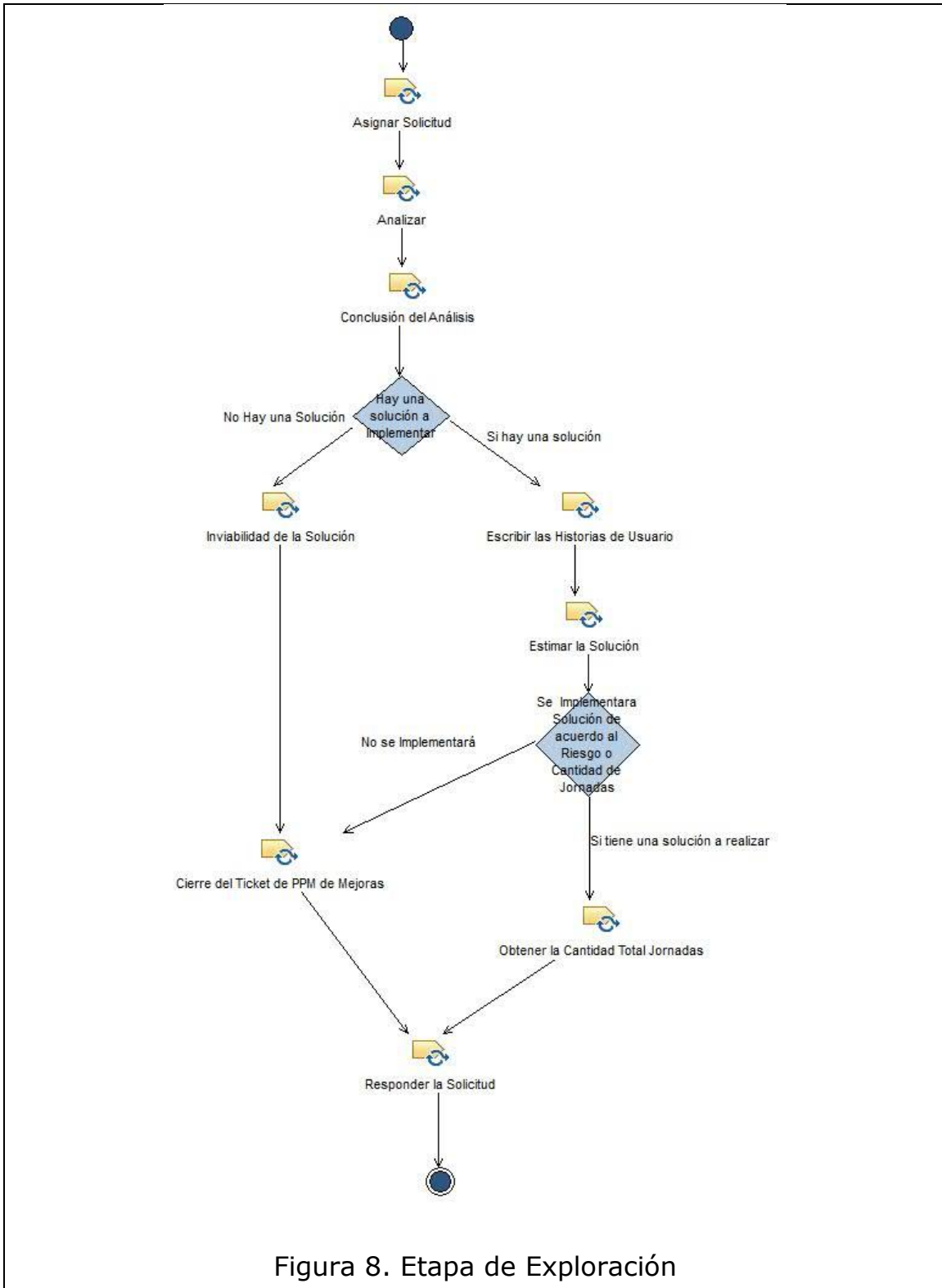


Figura 8. Etapa de Exploración

Propósito	Analizar una solicitud.
Descripción	Se analiza la solicitud para determinar si esta tiene una solución. Si no tiene una solución se envía un correo al Dueño del Producto con la explicación del caso, y si tiene una solución se realiza una estimación del esfuerzo que implica la solución a implementar.

Tabla 11 Analizar una solicitud.

Descripción de las tareas:

Descripción de la Tarea	Asignar Solicitud
Roles	<ul style="list-style-type: none"> <li>❖ Responsable del Equipo.</li> <li>❖ Gestor de Mejoras.</li> </ul>
Entradas	<ul style="list-style-type: none"> <li>❖ Backlog.</li> <li>❖ Planificación (Anexo A3).</li> </ul>
Salidas	Pendiente de Revisión (Anexo A2).
Descripción	El Responsable del equipo asigna un ticket que está en el backlog a un Gestor de Mejoras para su análisis. De esta manera el ticket sale del backlog y pasa a la sección de " <b>Pendientes</b> ", del tablero visual.
Pasos	<ul style="list-style-type: none"> <li>❖ Se selecciona un tema a analizar: Según la priorización dada por el Dueño del Producto, el Responsable del Equipo selecciona un tema para analizar del backlog.</li> <li>❖ Se selecciona un Gestor de Mejoras: El Responsable del Equipo selecciona un Gestor de Mejoras para que analice el tema. Se selecciona el que esté disponible de acuerdo a la planificación del equipo.</li> <li>❖ Se posiciona el post-it en la sección Pendiente: Se saca el post-it que representa el tema y se deja en la sección de pendientes para que</li> </ul>

	pueda ser tomado por el Gestor de Mejoras cuando llegue el momento de analizar el tema.
Práctica	Tablero visual o Kanban.

Tabla 12 Asignar Solicitud.

Descripción de la Tarea	Analizar
Roles	<ul style="list-style-type: none"> <li>❖ Gestor de Mejoras.</li> <li>❖ Dueño del Producto.</li> <li>❖ Experto en el Módulo.</li> </ul>
Entradas	PPM de Mejoras.
Salidas	Resultado de la Investigación: Se cuenta con la revisión de la solicitud. Este análisis puede significar que exista una solución o no. De existir una solución se debe tener todas las posibles soluciones a implementar. Esta información no es necesaria tenerla en un formato particular y la tiene en su poder el Gestor de Mejoras asignado.
Propósito	Ver la factibilidad de implementar una solución a una solicitud.
Descripción	Analizar una solicitud y ver posibles soluciones a implementar.
Pasos	<ul style="list-style-type: none"> <li>❖ Leer la información Registrada: Se busca en PPM el tique de mejoras, y se revisa la información registrada.</li> <li>❖ Consulta a un experto del módulo: Si se tiene dudas de cómo funciona el módulo, aplicaciones, servicios o productos que se verá afectado por la solicitud, se debe buscar algún experto en primera instancia al interior del equipo o sino fuera del equipo (en la misma compañía).</li> <li>❖ Consulta al Dueño del Producto: Si hay dudas sobre lo solicitado se debe canalizar con el</li> </ul>

	<p>Dueño del Producto la respuesta de estas consultas, o que él las pueda gestionar con quien corresponda.</p> <ul style="list-style-type: none"> <li>❖ Finalización de la Investigación: El Gestor de Mejoras asignado concluye si la solicitud tiene o no una solución a implementar por el equipo, y si tiene una solución cuales serían las opciones.</li> </ul>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 13 Analizar.

Descripción de la Tarea	Conclusión del Análisis
Roles	<ul style="list-style-type: none"> <li>❖ Responsable del Equipo.</li> <li>❖ Gestor de Mejoras.</li> <li>❖ Dueño del Producto.</li> <li>❖ Experto en el Módulo.</li> </ul>
Entradas	Resultado de la Investigación.
Salidas	Solución por Implementar: Después de la reunión de equipo para revisar las posibles soluciones, se cuenta con una respuesta sobre la solución a implementar para la solicitud.
Propósito	Tener la respuesta a la posible solución a implementar o la respuesta de la factibilidad técnica.
Descripción	Cuando el Gestor de Mejoras asignado a una solicitud termina el análisis, el responsable del Equipo debe llamar a una reunión de equipo, en donde se explica lo analizado, y las posibles soluciones o bien la inviabilidad de alguna solución. Entre todo el equipo se define la solución o cuál es la respuesta que se debe entregar al Dueño de Producto, si no es factible implementar alguna solución.

Pasos	<ul style="list-style-type: none"> <li>❖ Gestor de Mejoras informa del término del análisis: Cuando el Gestor de Mejora termina el análisis le informa al Responsable del Equipo de este hito.</li> <li>❖ Reunión de Equipo: El Responsable del Equipo, llama a reunión al equipo. Se puede invitar al Dueño del Producto y a los Expertos en los Módulos en caso de ser necesario.</li> <li>❖ Discusión del Análisis: <ul style="list-style-type: none"> <li>• El Gestor de Mejoras asignado expone el análisis que realizó y los participantes de la reunión conversan de la conclusión.</li> <li>• De tener una solución a implementar, se escoge la solución que se abordará.</li> <li>• De no tener una solución se acuerda el motivo del por qué no es factible alguna solución para así informarle esta respuesta al Dueño del Producto que realizó la solicitud.</li> </ul> </li> </ul>
-------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 14 Conclusión del Análisis.

Descripción de la Tarea	Escribir las Historias de Usuario
Roles	<ul style="list-style-type: none"> <li>❖ Responsable del Equipo.</li> <li>❖ Gestor de Mejoras.</li> </ul>
Entradas	Solución por Implementar.
Salidas	Documento de Historias de Usuario (Anexo A4). Se usará el acrónimo HU, para hablar de las Historias de Usuario.
Propósito	Contar con documentación de la solución a implementar.
Descripción	Se escribe en lenguaje del usuario (es decir, en lenguaje del Negocio) lo que se va a implementar como solución. Los criterios de aceptación y los

	objetivos de la solución se describirán de manera técnica y funcional en caso de ser necesario.
Pasos	<ul style="list-style-type: none"> <li>❖ Escribir nombre de la HU: Se asigna un nombre para la HU.</li> <li>❖ Se describe la solución: En lenguaje del usuario se describe cuál es la solución por implementar.</li> <li>❖ Escribir los Criterios de Aceptación: Se escribe cómo el usuario debería dar por aceptado el requerimiento. Puede ser más de un criterio y éstos pueden estar descritos de forma técnica (nombre de tablas, campos, de aplicaciones).</li> <li>❖ Escribir el Objetivo de la solución: Se escribe cuál es el objetivo de la solución, es decir, qué se va a lograr con esta solución.</li> </ul>
Práctica	Historias de Usuario.

Tabla 15 Escribir las Historias de Usuario.

Descripción de la Tarea	Estimar la Solución
Roles	<ul style="list-style-type: none"> <li>❖ Responsable del Equipo.</li> <li>❖ Gestor de Mejoras.</li> <li>❖ Dueño del Producto.</li> <li>❖ Experto en el Módulo.</li> </ul>
Entradas	Documento de Historias de Usuario.
Salidas	Estimación de la Solución.
Propósito	Estimar el esfuerzo de la Construcción (Anexo A5).
Descripción	Estimar en la Unidad Jornadas el esfuerzo que tomará la construcción de la implementación de la solicitud (Anexo A5). Y Asignar la Complejidad de la solución (Anexo A5).



Pasos	<ul style="list-style-type: none"> <li>❖ Se reúne al equipo: El responsable del equipo, cita a reunión de equipo.</li> <li>❖ Revisión de la historia de Usuario: Se revisa la Historia de Usuario. El Gestor de Mejoras que tiene a cargo el requerimiento lee la HU al equipo, y se realizan correcciones si fuese necesario.</li> <li>❖ Se asigna la complejidad al requerimiento: Cada integrante de la reunión, asigna la complejidad que él estima que tiene la solución. En consenso se asigna la complejidad final.</li> <li>❖ Se revisa las complejidades similares: Se revisan las estimaciones similares de otras solicitudes de acuerdo con la complejidad asignada a la solución.</li> <li>❖ Se estima la construcción: Con la una técnica adaptada de Planning Póker se estima el esfuerzo en jornadas de la construcción de la solución.</li> <li>❖ Se revisan los riesgos: El responsable del equipo consulta al equipo por los posibles riesgos de la implementación. Se anotan y se registran en el Kanban para tenerlos a la vista.</li> <li>❖ Se analiza la viabilidad de la implementación: El equipo analiza de acuerdo con la cantidad de jornadas y la complejidad la factibilidad de implementar la solución.</li> </ul>
Práctica	Planning Póker.

Tabla 16 Estimar la Solución.

Descripción de la Tarea	Obtener la Cantidad Total Jornadas
Roles	<ul style="list-style-type: none"> <li>❖ Responsable del Equipo.</li> <li>❖ Gestor de Mejoras.</li> <li>❖ Experto en el Módulo.</li> </ul>
Entradas	Estimación de la Solución.
Salidas	<ul style="list-style-type: none"> <li>❖ Documento de Estimación Final (Anexo A6).</li> <li>❖ Planificación.</li> </ul>
Propósito	Tener la cantidad total de jornadas (Anexo A6).
Descripción	Tener la cantidad total de jornadas a utilizar para la implementación de la solución del requerimiento.
Pasos	<ul style="list-style-type: none"> <li>❖ Reunión de Estimación Final: El Responsable del Equipo se reúne con el Gestor de Mejoras que realizó el análisis.</li> <li>❖ Llenado los ítems de la estimación: Se completa cada uno de los ítems a considerar para la estimación (esto también puede significar consultar al equipo, o a los Expertos en los Módulos, por cuál es la mejor cantidad de jornadas para cada uno de los ítems): <ul style="list-style-type: none"> <li>• Se asignan jornadas para la ambientación de la solución.</li> <li>• Se asignan jornadas para escribir los casos de prueba.</li> <li>• Se toman las jornadas de construcción de la tarea de estimación.</li> <li>• Se asignan jornadas para documentar la solución.</li> <li>• Se asignan jornadas para realizar inspecciones.</li> <li>• Se asignan jornadas para certificar la solución, se ejecutan los casos de prueba.</li> <li>• Se asignan jornadas como reserva ante imprevistos.</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>• Se asignan jornadas para versionar el cambio y piezas de software.</li> <li>❖ Pre planificación: se determinan las posibles iteraciones que se puede implementar la solución, se completa el calendario para determinar cuál es la fecha de inicio y término de cada iteración.</li> </ul>
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 17 Obtener la Cantidad Total Jornadas.

Descripción de la Tarea	Inviabilidad de la Solución
Roles	<ul style="list-style-type: none"> <li>❖ Responsable del Equipo.</li> <li>❖ Gestor de Mejoras.</li> </ul>
Entradas	Resultado de la Investigación: Se cuenta con la revisión de la solicitud. Este análisis puede establecer que exista una solución o no. De existir una solución se debe tener todas las posibles soluciones a implementar. Esta información no es necesario tenerla en un formato particular.
Salidas	PPM Mejoras.
Propósito	Tener registro del por qué no se implementará una solución para la solicitud.
Descripción	Según el resultado del análisis realizado, se registra esta decisión en PPM, en el ticket de Mejoras.
Pasos	<ul style="list-style-type: none"> <li>❖ Redacción del motivo: Se redacta por qué no se podrá implementar una solución.</li> <li>❖ Nota en PPM: Este motivo se registra como nota en el ticket de Mejoras en PPM.</li> </ul>

Tabla 18 Inviabilidad de la Solución.

Descripción de la Tarea	Cierre del Ticket de PPM de Mejoras
Roles	Responsable del Equipo.
Entradas	PPM Mejoras.
Salidas	PPM Mejoras.
Propósito	Dar por cerrada la solicitud.
Descripción	Se cierra el ticket de Mejoras en PPM, con el paso a producción de una implementación, o con su cancelación, por qué no se realizará la solicitud.
Pasos	<ul style="list-style-type: none"> <li>❖ Reunir la Información Relevante: Se reúne toda la información que se pueda del requerimiento y se respalda en el ticket de Mejoras en PPM.</li> <li>❖ Cerrar PPM Mejoras: Cerrar el ticket de Mejoras en PPM.</li> </ul>

Tabla 19 Cierre del Ticket de PPM de Mejoras.

Descripción de la Tarea	Responder la Solicitud
Roles	<ul style="list-style-type: none"> <li>❖ Responsable del Equipo.</li> <li>❖ Gestor de Mejoras.</li> <li>❖ Dueño del Producto.</li> </ul>
Entradas	<ul style="list-style-type: none"> <li>❖ Documento de Estimación Final.</li> <li>❖ Documento Historia Usuario.</li> <li>❖ Planificación.</li> <li>❖ Resultado de la Investigación.</li> <li>❖ Solución a Implementar.</li> </ul>
Salidas	<ul style="list-style-type: none"> <li>❖ PPM Mejoras.</li> <li>❖ Correo al Dueño del Producto con el resultado del Análisis.</li> </ul>

Propósito	Se responde al Dueño del Producto sobre la viabilidad de implementar una solución.
Descripción	Se responde al Dueño del Producto, si se implementará una solución o no. Si no se implementa una solución se informa el motivo. Si se realiza una solución, se informa cuál será, la cantidad de jornadas a consumir, las iteraciones y las fechas de inicio y entrega de cada iteración.
Pasos	<ul style="list-style-type: none"> <li>❖ Respaldo de información: La Historia de Usuario, la estimación final y la pre planificación se respaldan en PPM.</li> <li>❖ Se envía respuesta (un correo) al Dueño del Producto: La respuesta al Dueño del Producto sobre el resultado del proceso de exploración del requerimiento puede ser: <ol style="list-style-type: none"> <li>1. Del análisis, resulta que es inviable una solución al requerimiento. Se debe enviar este motivo al Dueño del Producto.</li> <li>2. De la estimación de la construcción, ya que puede que la implementación sea muy riesgosa, o sean muchas jornadas y no se realice alguna implementación. Esto ya que hay que implementar soluciones medianas y pequeñas (definido por la compañía menor a 45 jornadas de esfuerzo, esto para competir con el Área de Desarrollo). De lo contrario se debe indicar al Dueño del Producto que la solución debe ser vista por el Área de Desarrollo.</li> <li>3. La descripción de la solución, más la cantidad de jornadas a consumir y las fechas de inicio y termino.</li> </ol> </li> <li>❖ Respaldo de respuesta al Dueño del Producto: Se respalda en PPM la respuesta enviada al Dueño del Producto.</li> </ul>

Tabla 20 Responder la Solicitud.

### 3.2.5 Planificación

La actividad de Planificación consta de las tareas para ordenar el trabajo y las personas que participaran en la iteración. Como la compañía es una empresa proveedora de servicios TI, las iteraciones (para conseguir la implementación total de la solución), tienen una duración de acuerdo con lo que el Dueño del Producto, considere que es correcto para su realidad.

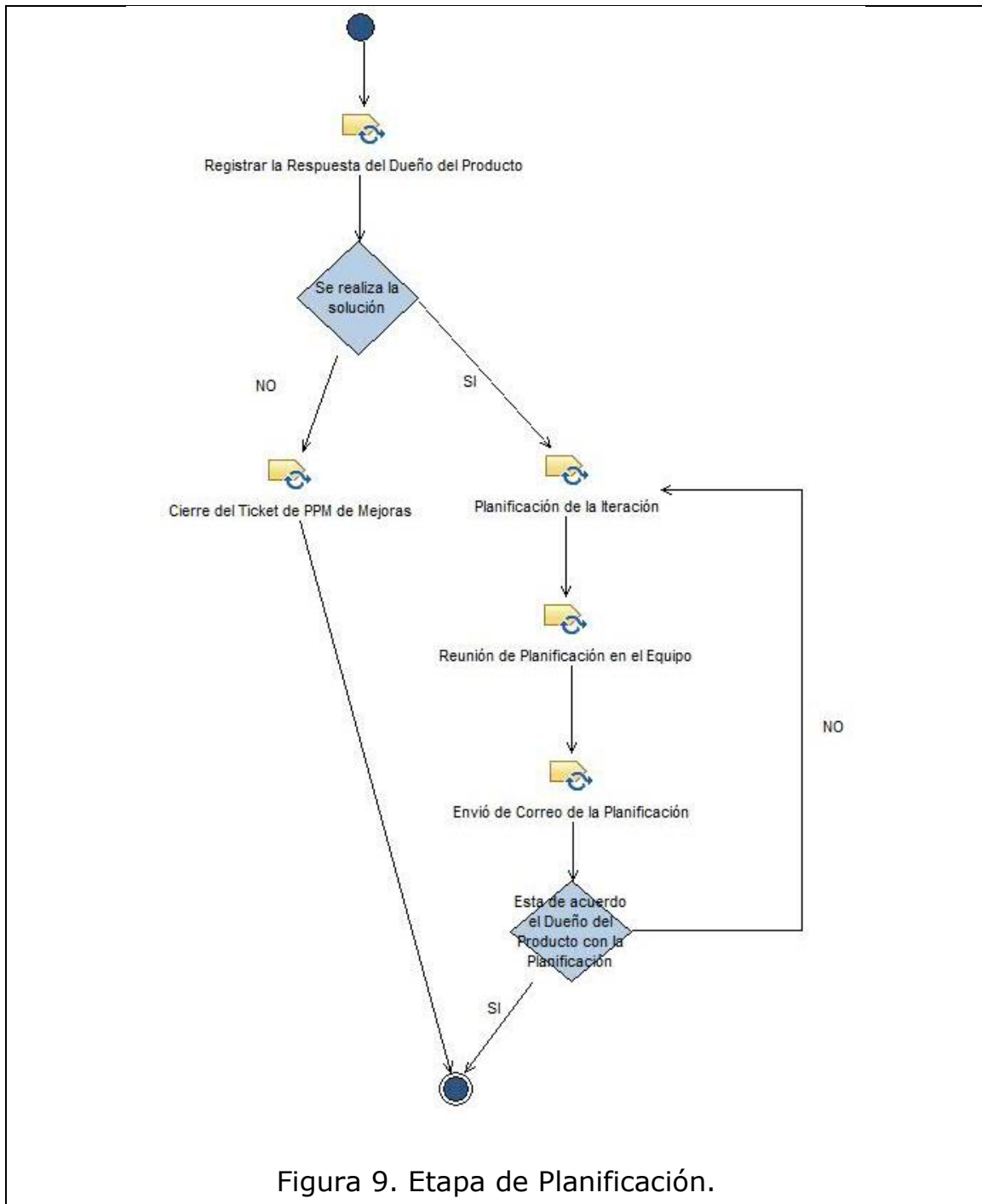


Figura 9. Etapa de Planificación.

Propósito	Planificar la ejecución de las iteraciones acordadas para la implementación de la solución de la solicitud.
Descripción	<p>Si el Dueño del Producto aprueba la implementación de la solicitud, entonces se planifica su ejecución. Si se puede cumplir con la planificación entregada en el presupuesto se tomará esa como la planificación final. Toda esta información se registra en el tique de Mejoras en PPM.</p> <p>Si el Dueño del Producto no aprueba la implementación entonces se solicita esta respuesta por correo con el motivo, se registra esta información en el ticket de Mejoras en PPM, se cierra la solicitud y se saca el post-it de la sección pendientes del tablero.</p>

Tabla 21 Etapa de Planificación.

Descripción de las tareas:

Descripción de la Tarea	Registrar la Respuesta del Dueño del Producto
Roles	<ul style="list-style-type: none"> <li>❖ Responsable del Equipo.</li> <li>❖ Dueño del Producto.</li> <li>❖ Gestor de Mejoras.</li> </ul>
Entradas	<p>Respuesta si se Implementa o no la Solución: Es un correo enviado por el Dueño del Producto sobre si se implementa o no la solución propuesta enviada. Este correo no tiene que tener un formato específico, pero debe dejar manifiesto si se realizará o no la implementación.</p> <p>Si no se realiza, este correo debe indicar por qué este no se realizará.</p>
Salidas	PPM Mejoras.

Propósito	Registrar en PPM la respuesta del Dueño del Producto sobre la realización de la implementación de la solución.
Descripción	Se ingresa como nota en PPM (en el ticket de Mejoras), la respuesta del Dueño del Producto sobre la realización de la implementación de la solución.
Pasos	<ul style="list-style-type: none"> <li>❖ Dueño del producto envía correo: El Dueño del Producto envía un correo al Responsable del Equipo sobre la decisión si se realiza o no la implementación de la solución propuesta.</li> <li>❖ Se Registra el correo en PPM: El Responsable del Equipo registra la respuesta en PPM como una nota en el ticket de Mejoras.</li> <li>❖ Se comunica al equipo de la decisión: El Responsable del Equipo (Gestores de Mejoras) comunica la respuesta del Dueño del Producto (puede ser en cualquier reunión o instancia, no necesariamente una reunión para comunicar esta decisión).</li> </ul>

Tabla 22 Registrar la Respuesta del Dueño del Producto.

Descripción de la Tarea	Planificación de las Iteraciones
Roles	<ul style="list-style-type: none"> <li>❖ Responsable del Equipo.</li> <li>❖ Dueño del Producto.</li> <li>❖ Gestor de Mejoras.</li> </ul>
Entradas	Documento de Estimación Final.
Salidas	Planificación.
Propósito	Definición de la planificación y asignación de tareas.



Descripción	Definición de la planificación, iteraciones y asignación de tareas. De esta manera se realiza un calendario con las tareas a realizar cada día.
Pasos	<ul style="list-style-type: none"> <li>❖ Definición de Iteraciones: se acuerda entre el Responsable del Equipo y el Dueño del producto las iteraciones a realizar, y las fechas de entrega de cada una. Según la estimación realizada en la etapa de análisis</li> <li>❖ Definición de Tareas: se llevan las actividades de la implementación de la solución a una tarea diaria.</li> <li>❖ Asignación de tareas: Se asignan las tareas a un responsable dentro de los Gestores de Mejoras.</li> </ul>
Practica	Diagrama Burndown.

Tabla 23 Planificación de las Iteraciones.

Descripción de la Tarea	Reunión de Planificación en el Equipo
Roles	<ul style="list-style-type: none"> <li>❖ Responsable del Equipo.</li> <li>❖ Gestor de Mejoras.</li> </ul>
Entradas	Planificación.
Salidas	<ul style="list-style-type: none"> <li>❖ Planificación.</li> <li>❖ PPM de Mejoras.</li> </ul>
Propósito	Que los involucrados en la implementación y ejecución de una iteración tengan conocimiento de su participación. Y el equipo en general tenga claro las fechas involucradas en la iteración.
Descripción	En una reunión de equipo queda definido cuál es la planificación, se asigna quién construye, quién prueba, quién inspecciona, quién certifica, considerando también las soluciones comprometidas y la carga del equipo.

Pasos	<ul style="list-style-type: none"> <li>❖ Revisión de la Planificación: El Responsable del Equipo, revisa la planificación de la iteración a ejecutar y la reorganiza según necesidad.</li> <li>❖ Reunión de Equipo: El Responsable del Equipo llama a reunión a los involucrados en la solución.</li> <li>❖ Se muestra la planificación al equipo: Se muestra la planificación a los involucrados en la implementación de la solución.</li> <li>❖ Adecuaciones a la planificación: Se realizan las adecuaciones necesarias a la planificación. Se tomará como base la planificación propuesta, pero se conversa de posibles situaciones del equipo como asignaciones o vacaciones.</li> <li>❖ Respaldo de la Planificación: El Responsable del Equipo respalda la planificación resultante en el ticket de PPM.</li> </ul>
Práctica	Planificación del Sprint de Scrum.

Tabla 24 Reunión de Planificación en el Equipo.

Descripción de la Tarea	Envío de Correo de la Planificación
Roles	<ul style="list-style-type: none"> <li>❖ Responsable del Equipo.</li> <li>❖ Dueño del Producto.</li> </ul>
Entradas	Planificación.
Salidas	<ul style="list-style-type: none"> <li>❖ Correo al Dueño del Producto: con este correo el Dueño del Producto tiene la información de cuándo se libera cada iteración. Hay que considerar que depende de dónde sea el Dueño del Producto es el nivel de detalle que se puede enviar. Dado que el Dueño del Producto puede ser de la compañía o puede ser del Cliente, este nivel de información a mostrar cambia.</li> </ul>

	❖ Actualización del Tablero en la sección Planificaciones (Anexo A2).
Propósito	Se envía un correo al Dueño del Producto la planificación.
Descripción	Se informa el Dueño del Producto de las fechas de inicio y término de las iteraciones.
Pasos	<ul style="list-style-type: none"> <li>❖ Envío de Correo al Dueño del Producto: Se envía un correo al Dueño del Producto con la planificación final de la implementación.</li> <li>❖ Se publica la planificación: Se deja en el tablero la planificación, siempre teniendo cuidado que exista una planificación por tema publicada.</li> </ul>
Práctica	Tablero visual o Kanban.

Tabla 25 Envío de Correo de la Planificación.

### 3.2.6 Construcción

La actividad de Construcción, conforman las tareas que componen la implementación de solución de la iteración, tiene el siguiente proceso:

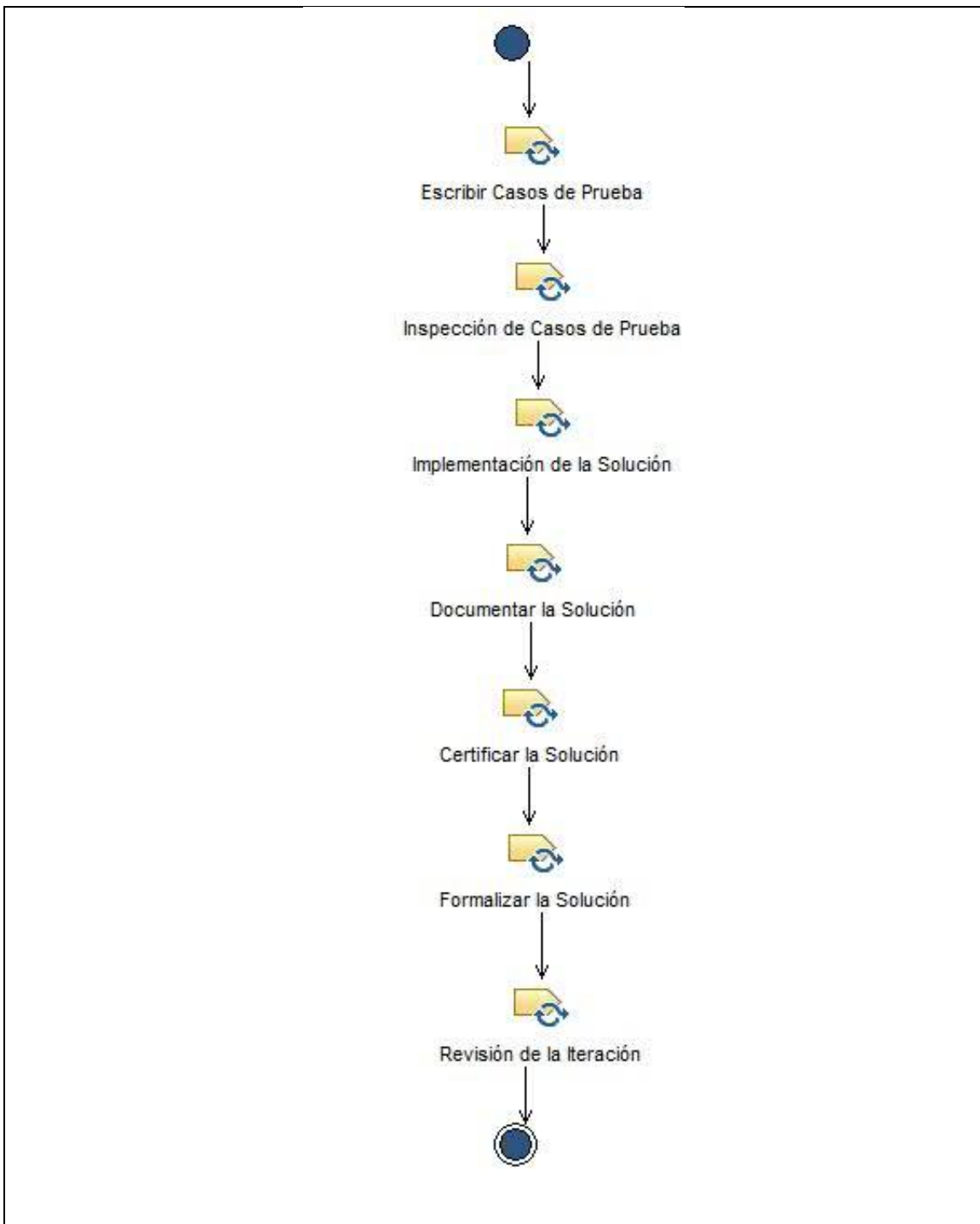


Figura 10. Etapa de Construcción de una Iteración.	
Propósito	Implementar la solución.
Descripción	En esta etapa se implementa una iteración, se certifica, se realizan inspecciones de código, y se crea el paquete a liberar.

Tabla 26 Etapa de Construcción de una Iteración.

Descripción de las tareas:

Descripción de la Tarea	Escribir Casos de Prueba
Roles	<ul style="list-style-type: none"> <li>❖ Responsable del Equipo.</li> <li>❖ Experto en el Módulo.</li> <li>❖ Gestor de Mejoras.</li> </ul>
Entradas	Documento Historia Usuario (Anexo A4).
Salidas	Documento Casos de Prueba (Anexo A7).
Propósito	Tener los casos de prueba definidos y escritos.
Descripción	Definir los casos de prueba. La persona que escribe los casos de prueba no es el mismo que implementa la solución.
Pasos	<ul style="list-style-type: none"> <li>❖ Redacción de casos de prueba: En general el que escribe los casos de prueba es quien analizó la solicitud, y también el que los escribe no puede codificar la solución.</li> <li>❖ Resolución de Dudas: Si hay cualquier duda sobre cómo probar o implementar la solución, se debe consultar al Experto en el Módulo.</li> </ul>
Práctica	Se utiliza la forma de trabajo de TDD, en donde primero antes de comenzar a escribir líneas de código se escriben primero los test, después código que cumpla el test. Aquí, aunque no usamos pruebas unitarias automáticas si antes

	de escribir cualquier línea de código primero se escriben los casos de prueba, que para nuestro caso se escriben mezclando aspectos funcionales y técnicos.
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 27 Escribir Casos de Prueba.

Descripción de la Tarea	Inspección de Casos de Prueba
Roles	<ul style="list-style-type: none"> <li>❖ Responsable del Equipo.</li> <li>❖ Gestor de Mejoras.</li> </ul>
Entradas	Documento Casos Prueba.
Salidas	<ul style="list-style-type: none"> <li>❖ Documento Casos Prueba.</li> <li>❖ PPM de Mejoras.</li> </ul>
Propósito	Tener el documento de Casos de Prueba revisado.
Descripción	Verificar que los casos de prueba son entendidos por los Gestores de Mejoras que implementará la solución. Ante cualquier diferencia se modificará el documento hasta que sea entendido por todos.
Pasos	<ul style="list-style-type: none"> <li>❖ Revisión de los Casos de Prueba: Las personas que implementarán la iteración, revisan los casos de prueba.</li> <li>❖ Reunión de Equipo: El Gestor de Mejoras que escribió los casos de prueba se reúne con los Gestores que implementará la iteración y con el Responsable del Equipo (y todas las personas del equipo que puedan asistir), para revisar el documento de casos de prueba.</li> <li>❖ Actualización Documento de Casos de Prueba: El Gestor de Mejoras que escribió los casos de prueba, actualiza el documento con las observaciones dadas por el equipo.</li> </ul>

	<ul style="list-style-type: none"> <li>❖ Respaldo del Documento: El Gestor que escribió los casos de prueba, sube el documento al ticket de mejoras en PPM.</li> </ul>
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 28 Inspección de Casos de Prueba.

Descripción de la Tarea	Implementación de la Solución
Roles	<ul style="list-style-type: none"> <li>❖ Responsable del Equipo.</li> <li>❖ Gestor de Mejoras.</li> </ul>
Entradas	Documento Casos de Prueba.
Salidas	Construcción de la Solución (se cuenta con la implementación de la iteración).
Propósito	Contar con la solución de la iteración.
Descripción	Se implementan las funcionalidades de la iteración.
Pasos	<ul style="list-style-type: none"> <li>❖ Construcción mínima: Se construye el mínimo software que cumpla con las pruebas descritas en el documento de Casos de Pruebas.</li> <li>❖ Pruebas Unitarias: Se realizan pruebas unitarias, las cuales pueden no ser documentadas, que prueba el software construido con respecto a los Casos de Prueba.</li> <li>❖ Refactorización: Cuando se tiene el software que cumpla con la prueba, se termina la implementación con la incorporación de las normas.</li> <li>❖ Propiedad Colectiva: Todos los días se sube el código a un SVN, para dejar el código disponible a todo el equipo.</li> <li>❖ Revisión del Código: Otro Gestor de Mejoras revisa el código implementado y realiza contribuciones. El objetivo de esta práctica es que otra persona sepa lo que se está</li> </ul>

	<p>construyendo. Todos los días se debe destinar aproximadamente 30 minutos para validar el código.</p>
Práctica	<p>Prácticas de XP:</p> <ul style="list-style-type: none"> <li>❖ TDD: se está en la parte que se construye el software mínimo que cumple con los casos de prueba.</li> <li>❖ Refactorización: después que se ejecutan las pruebas unitarias por el Gestor de Mejoras que está implementando la solución (las cuales son definidas el desarrollador para probar su código, estas no forman parte de la solución ni son documentadas, en cambio las pruebas que se utilizan en la certificación si son documentadas y pueden ser usadas en nuevos desarrollos), se normaliza el código para tener la versión definitiva que se debe certificar.</li> <li>❖ Propiedad Colectiva de los Fuentes.</li> <li>❖ Trabajo de pares. En este trabajo utilizamos este término como para hacer un paralelo con la práctica "Programación de a Pares". La diferencia aquí no es que estén dos personas al mismo tiempo implementado una solución en un PC, sino que una persona implementa la solución y otra persona inspecciona el código cuando este es subido al SVN.</li> <li>❖ Código Estándar.</li> </ul>

Tabla 29 Implementación de la Solución.



Descripción de la Tarea	Documentar la Solución
Roles	<ul style="list-style-type: none"> <li>❖ Responsable del Equipo.</li> <li>❖ Gestor de Mejoras.</li> </ul>
Entradas	Documento Historia de Usuario.
Salidas	Documento Análisis, Desarrollo y Pruebas (siglas ADP, Anexo A8).
Propósito	Documentar la solución Implementada.
Descripción	Se describe lo realizado como solución y otras consideraciones especiales, todo esto en términos del cliente ya que esta información es para uso del cliente.
Pasos	<ul style="list-style-type: none"> <li>❖ Solución Funcional: Se debe documentar la solución funcional en términos lo más cercano al lenguaje del usuario que se pueda.</li> <li>❖ Solución Técnica: Se debe documentar la solución técnica de la solución implementada.</li> <li>❖ Inspección: El Responsable del Equipo se debe asegurar que el Gestor de Mejoras que realizó la inspección de código revise el documento ADP, y realice las modificaciones necesarias para que este sea entendible.</li> </ul>

Tabla 30 Documentar la Solución.

Descripción de la Tarea	Certificar la Solución
Roles	<ul style="list-style-type: none"> <li>❖ Responsable del Equipo.</li> <li>❖ Gestor de Mejoras.</li> </ul>
Entradas	<ul style="list-style-type: none"> <li>❖ Construcción de la Solución.</li> <li>❖ Documento de Casos de Prueba.</li> </ul>
Salidas	<ul style="list-style-type: none"> <li>❖ Construcción de la Solución.</li> <li>❖ Documento de Casos de Prueba.</li> <li>❖ PPM Mejoras.</li> </ul>
Propósito	Ejecutar los Casos de Pruebas definidos para la iteración.
Descripción	Realizar pruebas formales de la implementación de la iteración que aseguren la calidad del entregable.
Pasos	<ul style="list-style-type: none"> <li>❖ Ejecución de Casos de Prueba: Se ejecutan los pasos de cada caso de prueba.</li> <li>❖ Correcciones a la Solución: Si se producen errores en las pruebas, el que las esté ejecutando tendrá que revisar si es o no un error del software. Si lo es, él mismo debe solucionar el error en la medida de lo posible.</li> <li>❖ Propiedad Colectiva: Todos los días se sube el código a un SVN, para dejar el código disponible a todo el equipo.</li> <li>❖ Evidencia de las Pruebas: En el mismo documento de Casos de Prueba, se documenta la evidencia de la ejecución de los pasos.</li> <li>❖ Respaldo de Documento Casos de Prueba: El Responsable del Equipo, sube el archivo de Casos de Prueba a PPM.</li> <li>❖ Inspecciones del Documento de Casos de Prueba: El Responsable del Equipo debe asegurar que la persona que implementó la solución lea la evidencia de las pruebas, y realice los cambios de ser necesario.</li> </ul>

Práctica	❖ Propiedad colectiva del software.
----------	-------------------------------------

Tabla 31 Certificar la Solución.

Descripción de la Tarea	Formalizar la Solución
Roles	<ul style="list-style-type: none"> <li>❖ Responsable del Equipo.</li> <li>❖ Gestor de Mejoras.</li> </ul>
Entradas	<ul style="list-style-type: none"> <li>❖ Construcción de la Solución.</li> <li>❖ Documento de Casos de Prueba.</li> <li>❖ Documento ADP.</li> </ul>
Salidas	Construcción de la Solución.
Propósito	Versionar los cambios en el repositorio oficial de la compañía.
Descripción	Se versionan las fuentes modificados, se realizan los instaladores, se redacta la pauta de instalación y de reversa que se enviarán al cliente.
Pasos	<ul style="list-style-type: none"> <li>❖ Solicitud de piezas: Si hay piezas de software a modificar, se solicitan a través de la herramienta de versionamiento, para así tener el permiso de modificación.</li> <li>❖ Versionamiento de piezas nuevas: Si hay nuevas piezas creadas en la implementación, estas son versionadas en la herramienta de Configuración.</li> <li>❖ Creación de Instaladores: Se crean los instaladores necesarios: <ul style="list-style-type: none"> <li>• BD.</li> <li>• Unix.</li> <li>• WEB.</li> <li>• etc.</li> </ul> </li> <li>❖ Se construye el entregable: Se adjunta el documento ADP, el Documento de Casos de</li> </ul>

	<p>Prueba, y los instaladores, para generar un entregable al cliente.</p> <ul style="list-style-type: none"> <li>❖ Se escribe la pauta de instalación: Se describen los pasos que el cliente deberá hacer para instalar la solución en sus ambientes, tanto de preproducción como producción.</li> </ul>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 32 Formalizar la Solución.

Descripción de la Tarea	Revisión de la Iteración
Roles	<ul style="list-style-type: none"> <li>❖ Responsable del Equipo.</li> <li>❖ Gestor de Mejoras.</li> </ul>
Entradas	Construcción de la Solución.
Salidas	Construcción de la Solución.
Propósito	Inspección del entregable a liberar al cliente.
Descripción	Se revisa el paquete que se enviará al cliente y se entrega este al cliente.
Pasos	<ul style="list-style-type: none"> <li>❖ Reunión de Equipo: El Responsable del Equipo, cita a reunión a todos miembros del equipo.</li> <li>❖ Se Revisa la Pauta de Instalación: Se revisan los pasos que se deben seguir para la instalación de la solución.</li> <li>❖ Se revisan los instaladores: Se revisan que los instaladores sigan la normativa tanto de nombre como de estructura.</li> <li>❖ Se revisa la Documentación: Se revisa que los documentos ADP y la evidencia de las pruebas estén en el paquete.</li> <li>❖ Se entrega el paquete: A través de correos se libera el paquete al cliente.</li> </ul>

Tabla 33 Revisión de la Iteración.

### 3.2.7 Producción

La actividad de Producción tiene el siguiente proceso:

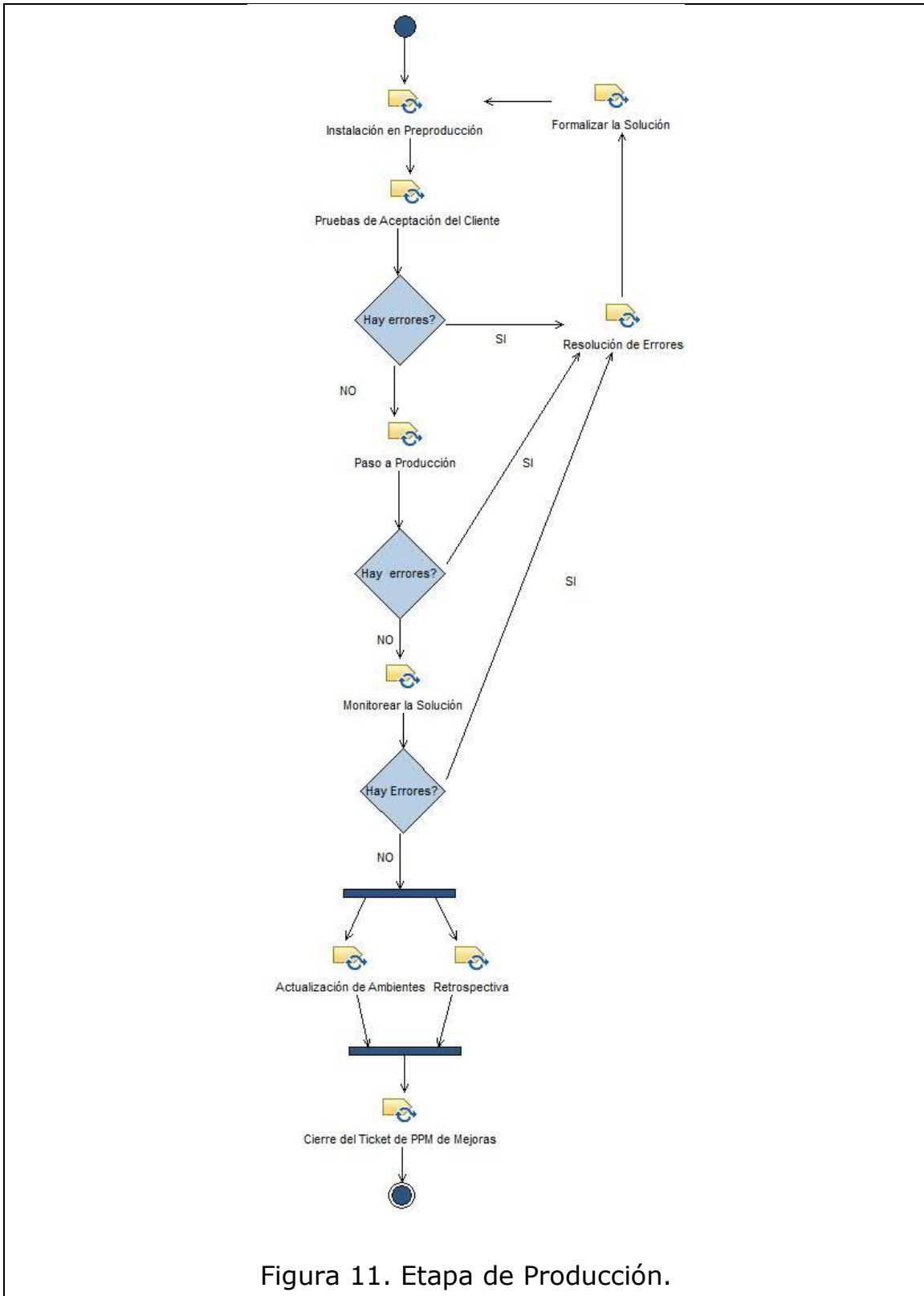


Figura 11. Etapa de Producción.

Propósito	Pasarse a producción la iteración implementada.
Descripción	<p>Para poder pasar la solución a producción, primero hay que pasar el entregable a preproducción para que el cliente pueda realizar las pruebas de aceptación.</p> <p>Si hay errores se deben resolver hasta que se superen estas pruebas. Se realiza el paso a producción y se monitorea un tiempo de acuerdo con lo acordado con el cliente. El Dueño del Producto, define en su planificación el tiempo que los Usuarios harán pruebas por lo que se designa una persona del equipo de Gestores para su atención. Si no es necesario que este toda la jornada esperando por si se levanta algún inconveniente se asigna un porcentaje de la jornada a esta revisión, y el Gestor de Mejoras realiza otras actividades en el resto de la jornada. Ahora si el Dueño del Producto no informa el tiempo que se harán las pruebas de aceptación, las observaciones quedan en el backlog a la espera de su priorización por parte del Dueño del Producto. Esta es la forma que la compañía siempre ha trabajado esta etapa con los clientes, esta iteración para la resolución de errores no es algo definido en este proceso.</p> <p>Una vez ya pasado el tiempo de monitoreo se actualizan los ambientes de producción de la compañía para que cualquier incidencia sea resuelta por Soporte Correctivo que realiza Gestión de Incidencias con los clientes.</p>

Tabla 34 Etapa de Producción.

## Descripción de las tareas:

Descripción de la Tarea	Instalación en Preproducción
Roles	<ul style="list-style-type: none"> <li>❖ Responsable del Equipo.</li> <li>❖ Gestor de Mejoras.</li> <li>❖ Dueño del Producto</li> </ul>
Entradas	Construcción de la Solución.
Salidas	Entregable Instalado en Preproducción: El resultado es tener un paquete entregable que sea instalado en ambientes de preproducción del cliente ya que es el mismo que se utilizará para el paso a producción. El ambiente de preproducción del cliente debe ser similar al ambiente de producción.
Propósito	Probar los instaladores y tener la solución instalada para la realización de pruebas de aceptación.
Descripción	<p>Instalar el entregable en los ambientes de pruebas del cliente, para así poder probar los instaladores en ambientes del cliente y poder pasar a realizar las pruebas de aceptación.</p> <p>Estas acciones son realizadas por el cliente o con quien el Dueño del Producto coordine.</p>
Pasos	<ul style="list-style-type: none"> <li>❖ Se genera ticket para paso a preproducción: El Dueño del Producto o quien coordine con él crea los tickets para el paso a preproducción del entregable.</li> <li>❖ Instalación de la solución: El cliente instala el entregable en preproducción.</li> <li>❖ Revisión del Paso a Preproducción: El Dueño del Producto debe revisar que el entregable quede correctamente instalado o también solicitar ayuda al Responsable del Equipo para revisar la instalación. Si hay errores se debe</li> </ul>

	armar nuevamente el paquete e instalar en preproducción.
--	----------------------------------------------------------

Tabla 35 Instalación en Preproducción.

Descripción de la Tarea	Pruebas de Aceptación del Cliente
Roles	<ul style="list-style-type: none"> <li>❖ Dueño del Producto.</li> <li>❖ Usuarios del Sistema.</li> <li>❖ Responsable del Equipo.</li> <li>❖ Gestor de Mejoras.</li> </ul>
Entradas	Entregable Instalado en Preproducción.
Salidas	Resultado de Pruebas de Aceptación: Los Usuarios del Sistema, documentan las pruebas en el formato que el cliente estime; si estas son todas satisfactorias se procede con el paso a producción. Si hay defectos estos deben ser enviados al Responsable del Equipo por el Dueño del Producto para que sean revisados.
Propósito	Los Usuarios del Sistema realizan un conjunto de pruebas para certificar la solución.
Descripción	El Dueño del Producto coordina con los usuarios la ejecución de Casos de Prueba para certificar la solución y si estas están correctas preparar el paso a producción. Los casos de prueba son definidos por el cliente (ya sea por el Dueño del Producto o los usuarios del sistema).
Pasos	<ul style="list-style-type: none"> <li>❖ Cliente prepara set de pruebas: El Usuario prepara un set de pruebas a certificar. El Dueño del Producto coordina con el Usuario y los usuarios este conjunto de pruebas.</li> <li>❖ El Usuario prepara datos de prueba: Los Usuarios del Sistema, preparan set de datos para realizar pruebas.</li> </ul>



	<ul style="list-style-type: none"> <li>❖ Se ejecutan casos de prueba: Los Usuarios del sistema ejecutan los casos de prueba. Estas pueden ser apoyadas por el equipo que implementó la solución. Para esto, el Dueño del Producto coordina con el Responsable del Equipo que algún Gestor de Mejoras esté presente en estas pruebas.</li> <li>❖ Revisión de Errores: el Dueño del Producto envía las pruebas al Responsable del Equipo para que los Gestores de Mejoras revisen si los defectos encontrados por los Usuarios son o no errores. Los que sean errores habrá que resolverlos y se tendrá que enviar nuevamente un entregable para que sea instalado y probado. En conjunto se define si hay errores o no. Si la observación se trata de un cambio de alcance no considerado en la iteración por el Dueño del Producto en esta iteración, se determina su envergadura y si esta no es de gran escala se hace como si fuera un error, es decir, se da atención casi inmediatamente. En cambio, si este tiene alguna envergadura mayor se debe revidar la planificación y pasa como una nueva iteración.</li> </ul>
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 36 Pruebas de Aceptación del Cliente.

Descripción de la Tarea	Resolución de Errores
Roles	<ul style="list-style-type: none"> <li>❖ Dueño del Producto.</li> <li>❖ Responsable del Equipo.</li> <li>❖ Gestor de Mejoras.</li> </ul>
Entradas	Resultado de Pruebas de Aceptación.
Salidas	Construcción de la Solución.
Propósito	Resolver los errores que resulten de las pruebas de aceptación.
Descripción	El equipo que implementó la solución resuelve los errores encontrados en la etapa de pruebas de aceptación.
Pasos	<ul style="list-style-type: none"> <li>❖ Se analiza la solución del error: Se analiza el error reportado y se busca la solución.</li> <li>❖ Se implementa la solución a los errores: Se construye la solución a los problemas presentados.</li> </ul>

Tabla 37 Resolución de Errores.

Descripción de la Tarea	Paso a Producción
Roles	<ul style="list-style-type: none"> <li>❖ Dueño del Producto.</li> <li>❖ Responsable del Equipo.</li> <li>❖ Gestor de Mejoras.</li> <li>❖ Usuarios del Sistema</li> </ul>
Entradas	Construcción de la Solución.
Salidas	Solución en Producción: Dejar operativa la implementación de la solución. Se realizan algunas pruebas de continuidad operacional de las aplicaciones. El Dueño del Producto envía un correo con el resultado del paso a producción.

Propósito	Instalar la solución en el ambiente de producción.
Descripción	El Dueño del Producto o con quien él acuerde en el Cliente, generará los tickets de paso a producción. Se instala el paquete en producción y el cliente realiza pruebas de regresión para comprobar que la aplicación funciona.
Pasos	<ul style="list-style-type: none"> <li>❖ Generación de ticket de cambio: El Dueño del Producto o con quien él coordine, genera los tickets de cambios necesarios para el paso a producción.</li> <li>❖ Instalación en Producción: Se instala en producción la solución.</li> <li>❖ Pruebas de Regresión: Los usuarios realizan pruebas para comprobar que las aplicaciones funcionan.</li> <li>❖ Resultado del Paso a Producción: El Dueño del Producto envía un correo con el resultado del paso a producción. Si hay errores da la orden de reversa del paso a producción y el Responsable del Equipo debe revisar estos errores y solucionarlos.</li> </ul>

Tabla 38 Paso a Producción.

Descripción de la Tarea	Monitorear la Solución
Roles	<ul style="list-style-type: none"> <li>❖ Dueño del Producto.</li> <li>❖ Responsable del Equipo.</li> <li>❖ Gestor de Mejoras.</li> </ul>
Entradas	Solución en Producción.
Salidas	Revisión del Monitoreo en Producción: Se responde al Cliente por el resultado del monitoreo. Esto se realiza a través de un correo que se envía al Dueño del Producto. Si hay errores se deben resolver y volver a pasarlos a producción.
Propósito	Se revisa el funcionamiento de la solución.
Descripción	Se monitorea el funcionamiento de la solución en producción y se hacen las correcciones correspondientes en caso de ser necesario.
Pasos	<ul style="list-style-type: none"> <li>❖ Se revisa el funcionamiento de la solución: Se revisan logs, modificaciones en la BD, o cualquier dato que permita determinar cómo está funcionando la solución.</li> <li>❖ Se responde al Dueño del Producto: Se envía un correo al Cliente sobre el resultado del monitoreo. Si hay errores, éstos se deben resolver con un nuevo paso a producción. El Dueño del Producto debe decidir si reversa el paso a producción, que dependerá de la gravedad del error.</li> </ul>

Tabla 39 Monitorear la Solución.

Descripción de la Tarea	Actualización de Ambientes
Roles	<ul style="list-style-type: none"> <li>❖ Responsable del Equipo.</li> <li>❖ Gestor de Mejoras.</li> </ul>
Entradas	Revisión del Monitoreo en Producción.
Salidas	<p>Ambientes Internos Actualizados: Se actualizan los ambientes de producción internos de la compañía para que, de haber problemas con la aplicación, ésta sea resuelta por Soporte Correctivo. O si estos requieren ser tomados por algún proyecto, ya con este paso estarán disponibles para todos.</p> <p>Mientras el código fuente este en esta etapa de desarrollo, estos están versionados en una herramienta llamada Dimensions, la cual permite si se trata de una modificación de una aplicación, tener una línea del software en producción y una del desarrollo. Si el mismo cliente solicita a la compañía algún cambio a otro equipo de desarrollo por medio de otro Dueño del Producto, con esta herramienta se ve las modificaciones que tiene la aplicación, y con dependiendo de la urgencia del nuevo cambio determinar junto con el Dueño del Producto, si el nuevo desarrollo contiene las modificaciones que aún no pasan a producción o debe ser otra línea de modificación de la aplicación que no se contendrán la una a la otra, por lo que pasando una a producción la otra modificación tendrá homologaciones pendientes.</p> <p>El Dueño del Producto, debe velar porque las aplicaciones no sean modificadas en producción durante el periodo de pruebas congelando en el cliente las incidencias de las aplicaciones involucradas, si hay incidencias en producción urgentes que requieren una modificación de la aplicación se debe entonces planificar la</p>

	homologación de la solución con respecto a la de producción como una nueva iteración o como una modificación en la etapa de pruebas de aceptación. Esto es definido por el equipo de desarrollo.
Propósito	Se actualizan los ambientes de producción internos, ya que en el cliente la solución ya se encuentra en producción.
Descripción	Se actualiza el repositorio de piezas de software de producción con la solución instalada en producción. Con este ítem, la implementación se encuentra totalmente en producción.
Pasos	<ul style="list-style-type: none"> <li>❖ Se crea la solicitud de actualización de ambientes: El responsable del Equipo debe asegurarse que algún Gestor de Mejoras genere una solicitud de actualización de ambientes al área de Configuración.</li> <li>❖ Se revisa la instalación: Se revisa la actualización de repositorios; si hay errores, éste se debe rechazar y volver a revisar hasta que esté correcta.</li> </ul>

Tabla 40 Actualización de Ambientes.

Descripción de la Tarea	Retrospectiva
Roles	<ul style="list-style-type: none"> <li>❖ Responsable del Equipo.</li> <li>❖ Gestor de Mejoras.</li> </ul>
Entradas	Revisión del Monitoreo en Producción.
Salidas	PPM de Mejoras.
Propósito	Los participantes de la implementación revisan el trabajo realizado.
Descripción	El equipo evalúa los puntos positivos que se realizaron, los negativos y qué acción concreta se realizará para mejorar los puntos negativos. A veces se juntan las retrospectiva de varias iteraciones.
Pasos	<ul style="list-style-type: none"> <li>❖ Reunión de Equipo: El Responsable del Equipo llama a una reunión de equipo.</li> <li>❖ Se revisan los puntos positivos: Cada participante describe los puntos positivos y qué se va a mantener.</li> <li>❖ Se revisan los puntos negativos: Cada participante describe los puntos negativos que ocurrieron en la iteración, y cuáles serían las acciones concretas para mejorarlos.</li> <li>❖ Se actualiza el ticket de mejoras: Con la conclusión se actualiza el ticket de PPM de mejoras.</li> </ul>
Práctica	<p>Retrospectiva.</p> <p>Mapas de Afinidad.</p>

Tabla 41 Retrospectiva.

# 4

## 4 Ejecución del Proceso

---

Como se describió en la sección anterior se definió un proceso inicial y una versión para el equipo de Soporte Mejoras, incorporando prácticas ágiles. El primer proceso se utilizó desde el 2013 al 2014 (con 137 iteraciones de soluciones), y la adaptación desde el 2015 hasta el primer semestre del 2016 (con 86 iteraciones de soluciones), que es hasta donde se muestran datos en este trabajo. Este proceso se utilizó en todas las solicitudes que fueron ingresadas al equipo.

Se usaron algunos indicadores, para ir evaluando el trabajo realizado por el equipo, entre estos:

- ❖ Cumplimiento: se mide la fecha de compromiso de la entrega de una iteración. Se cumple o no con la fecha comprometida.
- ❖ Calidad: se mide con los rechazos de las liberaciones entregadas al Dueño del Producto. Es decir, cuando una iteración es entregada, el Dueño del Producto coordina las pruebas con los usuarios, si hay rechazos estos son los que consideraran para medir la calidad.

En este capítulo se describe cómo se interactúa con el cliente y cómo resultó la aplicación del proceso en el trabajo del equipo.



## 4.1 Envíos hacia el Cliente

Para poder enviar los paquetes liberados a los Dueños del Producto, se utiliza PPM. La mecánica e hitos de estas iteraciones con el cliente se muestran en la siguiente figura:



Figura 12. Iteraciones entre Soporte Mejoras y los Dueños del Producto por PPM.

Cuando se crea un ticket en PPM para revisar una solicitud, se puede iterar con el cliente.

Iteración	Descripción
Solicitud de Información	Si se requiere información por parte del Dueño del Producto, se puede enviar esta solicitud a través del ticket de PPM. Se devuelve el ticket a Soporte Mejoras, cuando el Dueño del Producto responde la solicitud de información.
Envío de Respuesta	Es cuando se formaliza una respuesta al Dueño del Producto. Esta no implica un paquete de solución de software, sino que es un texto.
Envío de paquete de solución	Es cuando se envía al Dueño del Producto, un paquete de solución de software.
Se rechaza la entrega	El Dueño del Producto, puede rechazar una entrega, ya sea porque ésta vino de una respuesta de parte del equipo o es el envío de un paquete. Puede ser o bien porque no está de acuerdo con la respuesta enviada por el equipo, porque el paquete de software va con errores, o bien porque quiere agregar nuevos cambios al paquete antes de pasarlo a producción.
Se acepta rechazo	Es cuando el equipo acepta lo indicado por el Dueño del Producto. En general cuando se trata de un rechazo de un entregable es porque éste tiene errores. Si este rechazo se acepta, es que lo enviado va con defectos y hay que volver a enviar el paquete, por lo que esta aceptación de rechazo implica que la calidad del entregable no fue buena.
No se acepta rechazo	Es porque no se está de acuerdo ante un rechazo del Dueño del Producto, y se devuelve el ticket indicando los motivos de por qué no se está de acuerdo.

Tabla 42 Iteraciones con el Cliente en PPM.

Estas son las iteraciones cuando un ticket se envía por primera vez ya sea con un paquete o una respuesta. Con esto se cumple, el hito de la primera entrega formal. Después del primer envío el Dueño del Producto, tiene dos alternativas:

- ❖ Aprueba el envío: ya sea porque en el caso de enviarle una respuesta el Dueño del Producto, está de acuerdo o si es un entregable se realizaron las pruebas y la iteración sigue su paso a producción.
- ❖ Rechazar el envío: ya que puede que no esté de acuerdo con la respuesta o si se trata de un entregable éste en las pruebas tiene errores que hay que analizar.

En el caso de un rechazo, el equipo tiene las siguientes alternativas de acción:

- ❖ No aceptar el rechazo, ya que no se está de acuerdo con el rechazo del Dueño del Producto lo cual implica un nuevo hito con una respuesta formal.
- ❖ Se acepta el rechazo: es decir, lo indicado por el Dueño del Producto es correcto y el paquete de software liberado en la iteración debe ser enviado nuevamente. Esto representa un error en el envío. Hay oportunidades en que el Dueño del Producto necesita que en el mismo paquete que se está iterando se vean ajustes a lo enviado o acordado por lo que solicita que se haga un nuevo envío dentro del mismo tema, que no es un error a lo enviado. Para poder ser enviado este cambio, el Dueño del Producto debe rechazar el envío, pero este envío será marcado como un "Rechazo Aceptado No Contabilizado", el cual no afectará a la Calidad.

De estas iteraciones, las que significan un Hito son las que se está enviando una respuesta o un paquete de solución a la solicitud, estas son llamadas Envíos.

Cuando se rechaza el primer envío, se entra a la fase del segundo envío, el cual puede tener las mismas iteraciones del primer envío, más las iteraciones de aceptar el rechazo o no aceptar el rechazo, así el ciclo puede seguir hasta que el Dueño del Producto acepte y cierre el ticket.

## 4.2 Tasa de Cumplimiento

Tiene como objetivo medir el cumplimiento de los envíos de manera semestral, pero con el tiempo se dejó de medir ya que el primer año de medición el cumplimiento fue del 100%. En general son muy pocos los atrasos que se tienen y cuando los hemos tenido sólo han sido por poco tiempo y en general muy justificados.

La tasa de cumplimiento fue tomada de los indicadores de Gestión de Incidencias con la diferencia del período de medición que en este caso es semestral y en el caso de Gestión de Incidencias es mensual y, por lo tanto, para este proceso, el indicador es nuevo. La tasa de cumplimiento comenzó a medirse desde el 2013 con el inicio del primer proceso del equipo ya que éste se creó en septiembre del 2012, y utilizó este indicador desde el primer semestre del 2013. Se usó sólo el año 2013 y no se volvió a medir ya que las fechas de compromiso de las entregas liberadas se cumplían.

El indicador se medía de la siguiente manera. Cuando se planifica una iteración y se informa una fecha de entrega al Dueño del Producto, se mide si al final se cumplió o no esta fecha de compromiso. En un semestre, el universo total son todos los primeros envíos que se realizan. De estos envíos, algunos **SI** se cumplen y otros **NO** se cumplen. Este indicador es el porcentaje de los envíos que SI cumplen sobre el total de primeros envíos del semestre.

$$Tasa\ de\ Cumplimiento = \frac{\sum SI\ Cumple}{\sum SI\ Cumple + \sum NO\ Cumple}$$

Pero, aunque se estaba liberando en fecha los entregables, en un inicio la calidad de estos era pésima, teniendo constantes rechazos.

### 4.3 Calidad

La calidad de los entregables se define, para efectos de este trabajo, en base a los rechazos que nos hace de estos el Dueño del Producto. Este es el principal indicador que se define para el trabajo realizado por el equipo con las soluciones de software que se entregan. La fórmula de este indicador es:

$$Tasa\ de\ Rechazo = 100 \times \frac{\sum Rech\ Acep}{\sum Primeros\ Envios + \sum Rech\ Acep + \sum Rech\ Acep\ No\ Contab}$$

La tasa de rechazo es el porcentaje de los Rechazos Aceptados, versus los envíos válidos multiplicados por 100. Los envíos válidos para este indicador son los primeros envíos, los rechazos aceptados y los rechazos aceptados no contabilizados.

Los "**Rechazos No Aceptados**" no son considerados un envío válido para este indicador, ya que podría fomentar que, ante cualquier rechazo del Dueño del Producto, el equipo pueda No Aceptar el Rechazo, aunque este no corresponda con el fin de disminuir la Tasa de Rechazo.

La medición de este indicador se hace de manera semestral es decir de enero a junio una medición y de julio a diciembre de un año otra medición. Serán considerados los envíos válidos de este indicador en un período u otro de acuerdo a lo siguiente:

Envío	Periodo en donde se Aplica.
Primer Envío	Se aplica en el período en que se realizó el envío. Por ejemplo, si el envío se realizó el 3 de marzo, entonces este primer envío se aplica en el indicador del primer semestre.
Rechazo Aceptado y Rechazo Aceptado no Contabilizado	Aplica cuando al período al cual el cliente rechazó el requerimiento.

Tabla 43 Período donde se Aplica el Envío.

Desde el primer semestre del año 2013 y hasta el primer semestre del año 2016, se tiene la siguiente estadística:

Ítem	Valor
Cantidad de temas analizados en el período	305
Cantidad total de iteraciones implementadas en el período	223
Cantidad de iteraciones implementadas desde 2015, con el proceso actualizado.	86

Tabla 44 Cantidad de Iteraciones Realizadas.

Desde el primer semestre del año 2013 y hasta el primer del año 2016, se han analizado 305 solicitudes, y se han implementado 223 soluciones. De estas 223 soluciones, y con el proceso descrito en el capítulo 3, es decir desde el primer semestre del 2015, se han implementado 86 soluciones.

Según lo definido por la dirección de la empresa para el equipo de Soporte Mejoras, este indicador se da como cumplido si la Tasa de Rechazo, es menor o igual al 15%, si está entre 15 y 20% se considera como aceptable, y si mayor a 20% se considera como un mal resultado.

La Tasa de Rechazo desde el primer semestre del 2013 al segundo semestre del 2016 se resume en la siguiente tabla.

Semestre	Valor
1er Semestre 2013	26,2%
2do Semestre 2013	40%
1er Semestre 2014	40,7%
2do Semestre 2014	26%
1er Semestre 2015	19,6%
2do Semestre 2015	18%
1er Semestre 2016	8,3%

Tabla 45 Resultados de la Tasa de Rechazo en el Período de Trabajo.

El proceso descrito en el capítulo 3 se comenzó a aplicar en el año 2015. Puede verse que darle una estructura y mejoras al trabajo realizado ha resultado en una mejora sostenida de la tasa de calidad.

#### 4.4 Evaluación Cualitativa de la Incorporación de Prácticas Ágiles

En este capítulo se expone la evaluación que el equipo de Soporte Mejoras realizó a cada una de las prácticas de Scrum y XP que se utilizaron en el trabajo realizado.

De la revisión acerca de las prácticas ágiles que propone XP, tenemos el siguiente resultado:

## Evaluación Cualitativa de las Prácticas XP

### PRÁCTICA #4: Diseño Simple

Ante las constantes opiniones de los equipos de Soporte Correctivo sobre cómo se realizó la arquitectura de las aplicaciones WEB desarrolladas en la compañía, en Soporte Mejoras se han revisado las normas de arquitecturas y codificación existentes y estas se han cambiado para que las aplicaciones WEB sean más fáciles de montar. Antes para desplegar una aplicación y estructurar los fuentes requiere un esfuerzo muy considerable, ya que fueron construidas con mucho nivel de capas.

Se usaron herramientas para la compilación automática de las aplicaciones (como Maven), se utilizaron otros patrones de diseño que en la compañía no se utilizaban (como inyección de dependencias). Se hicieron reuniones con el personal de la compañía de distintas áreas para que nos dieran su opinión, teniendo estas actividades muy buena recepción.

No tuvimos complicaciones con la compañía sobre el cambio de los estándares ya que era una actividad que era necesaria realizar y que ningún equipo había realizado hasta entonces. Cuando se generaron las antiguas normas, existía un equipo dentro de la compañía que velaba por éstas, pero el equipo fue desarmado quedando sin una actualización de las normas.

### PRÁCTICA #5: Refactorización

Esta práctica se adoptó con buenos resultados y con muy buena acogida por parte del equipo. Se implementó de la siguiente forma:

- ❖ Primero se construye una solución mínima que cumpla con las pruebas.
- ❖ Después que esta funcionalidad está terminada y cumple con las pruebas unitarias ejecutadas por el desarrollador, se procede a aplicar las normas de codificación establecida (variables band, registro de trazas, log, normas, etc.).

Esta práctica también depende de las habilidades de cada persona ya que se puede escribir el código y dejarlo con las normas directamente sin necesidad de Re factorizar.



La incorporación de esta práctica se puede ver en la tarea "Implementación de la Solución", de la etapa de Construcción.

#### PRÁCTICA #6: Programación de a Pares

Esta práctica no se pudo poner en funcionamiento completamente por temas de imagen ya que los directivos de la empresa lo consideraban pérdida de personas que pueden hacer más cosas. Lo que sí se implementó es que, si un Gestor de Mejoras codifica una solución, otro Gestor asignado realiza una inspección de código y puede hacer cambios que se conversan al interior del equipo.

Para esto el que codifica publica el código para que al final del día sea otra persona quien lo revise.

Esto tuvo buenos resultados ya que contábamos con más de una persona que conocía el código además de la funcionalidad a implementar, lo que permite tener un backup. Ante posibles licencias o necesidad de tener más personas preparadas.

Cuando se certificaron estas soluciones se tuvo menos errores, ya que se compartieron ideas con la persona que inspeccionó el código.

#### PRÁCTICA #7: Entregas Cortas

Buscamos entregar soluciones en el tiempo más corto posible siempre asegurando calidad a los entregables. En general cuando los Dueños del Producto reciben las soluciones ellos deben buscar en sus usuarios quien certifique la solución, antes de pasarla a producción. Ahora como los usuarios siempre tienen poco tiempo, para los Dueños del Producto se les complica que sean muchas iteraciones porque se extiende la necesidad de contar con los usuarios.

Por lo que hay que estar en constante negociación con los Dueños del Producto, para poder tener la mayor cantidad de iteraciones posibles, ya que con esto se logra tener pasos más pequeños y permite ir reduciendo los riesgos. La compañía es una empresa proveedora de servicios para un Cliente, por lo tanto, la estrategia definitiva de las iteraciones depende del cliente y no de la compañía. Ya que el cliente tiene su proceso para pasar a producción las soluciones solicitadas por lo que bajo su planificación y recursos se acuerdan las iteraciones y los tiempos que ellos estimen más convenientes.

## PRÁCTICA #8: Pruebas

En esta práctica se propone el uso de TDD. Durante el año 2015 se revisó DreamWorks para poder realizar UnitTest y automatizar las pruebas unitarias, pero nos centramos en las pruebas unitarias en Java.

Las aplicaciones de la compañía están basadas en los siguientes lenguajes de programación Visual Basic 6.0, PL/SQL, pro\*c, .Net, lua, perl, por lo tanto, se dejó para un futuro la búsqueda de nuevos frameworks para el resto de los lenguajes.

Se realizaron pruebas unitarias en tres implementaciones en java, los fueron positivos, en lo referente a ayudar a ejecutar las pruebas de manera más rápida, pero el tiempo en programar las pruebas fue considerable. Además, en general en el corto plazo las aplicaciones modificadas o creadas no son nuevamente adaptadas por algún requerimiento, por lo que las pruebas automáticas generalmente fueron para probar la solución a implementar y no para tener pruebas para futuras implementaciones.

Codificar pruebas unitarias automáticas, por el motivo anteriormente expuesto fue desechado por el equipo por que se invertiría mucho tiempo en generar pruebas automáticas y por este momento no nos fue necesario invertir en este tipo de prácticas dejándolas para retomarla en algún momento. Pero lo que se necesitaba en ese momento era tener soluciones lo más rápido posible.

El aspecto que funciona es escribir las pruebas antes de codificar la solución. En el Área de Desarrollo de la compañía (que era la única área que realizaba proyectos evolutivos) se documentaba un extenso Análisis y Diseño de la solución, se construía y se documentaba un plan de pruebas para la certificación posterior a la construcción. Lo que nosotros hicimos fue juntar el documento de análisis y diseño junto con un plan de pruebas, en un solo documento, ya que la documentación de los casos de prueba debe ser técnica que oriente al desarrollador a construir la solución.

De esta manera el desarrollador codifica una solución para cumplir una prueba que está escrita. Y la certificación la ejecuta buscando

evidencia de que la funcionalidad se cumple y además se envía a los clientes como documentación de pruebas ejecutadas.

Así seguimos la forma de trabajo de TDD (pero las pruebas no son automáticas), ya que lo primero que tenemos es las pruebas, después se codifica y se realizan pruebas unitarias, se refactoriza y después se certifica. La certificación es realizada por algún miembro del equipo que no haya codificado la solución.

#### PRÁCTICA #9: Código Estándar

Hemos actualizado las normas de arquitectura y codificación, que tiene la empresa lo cual ha servido para estandarizar el código fuente.

La arquitectura que mejor resultados nos dio fue cambiar a arquitecturas de aplicaciones Java, basadas en EJB y hemos usado Inyección de Dependencias. Además, para ambientar las aplicaciones en Java se incorporó Maven. Con esto se logró generar ear's en minutos, siendo que con los antiguos servicios lograr ambientarlos desde cero era cosa de días. Además, logramos que los pesos de los ear's generados pasaran de megas a kilos, ya que los jar quedaron sin duplicar, problema que con las antiguas generaciones de ear's al estar los jar repetidos al interior del instalador aumentaba sin necesidad el peso de los ear's.

Esto ha dado muy buenos resultados ya que las aplicaciones WEB ahora son más livianas y sencillas.

Este tema nos ha ayudado siempre a estar buscando distintas innovaciones que podamos incorporar a la compañía.

#### PRÁCTICA #10: Propiedad Colectiva

Siguiendo la idea dada en la programación de a pares, el tema de designar a un Gestor de Mejoras que inspeccione el código de un compañero que está implementando la solución hizo que otra persona tuviera conocimiento de lo que se estaba implementado. Además, nos dio un respaldo si el Gestor de Mejoras que implementa la solución sufriera algún inconveniente para continuar con la implementación de la solución.

Además, antes del paso a producción de una solución se realizaron reuniones con los equipos de Soporte Correctivo que tendrían que

mantener el software modificado para que conocieran la solución tanto funcional como técnicamente.

Para alojar los cambios que se realizan previamente a formalizar la solución, se montó un servidor SVN. De esta manera el Gestor de Mejoras que realice la inspección de código puede buscar el código en el servidor y dejar los cambios que él realice en el mismo servidor.

Todos los días, el que esté implementando código dejará los fuentes en el SVN una hora antes de dejar de trabajar en donde serán tomados por el Gestor de Mejoras que realiza la inspección o por otra persona del equipo. Una vez que las piezas de software son versionadas en el repositorio oficial de la compañía, la solución es sacada del SVN.

La parte más complicada de la forma en que utilizamos la propiedad colectiva del software es que a veces a las personas se les olvida subir los fuentes de lo que se está desarrollando.

#### PRÁCTICA #11: Integración Continua

Esta práctica sí se utilizó inicialmente, pero luego se desechó. Para esto se implementó, como se indicó en la práctica anterior, un servidor SVN para que el código pueda ser visto e inspeccionado por los miembros del equipo de desarrollo.

El SVN permite, además de compartir el código, tener un respaldo de los cambios que se están trabajando.

Se realizó pruebas utilizando la aplicación Jenkins, pero en general, cuando se requería modificar una aplicación, siempre tenemos una iteración a la vez de esta aplicación, no tenemos distintos requerimientos que modifiquen la misma aplicación, que requieran que se estén integrando continuamente. Por lo tanto, quien esté implementando la solución es quien genera el componente que se entregará al Dueño del Producto. Por lo tanto, aunque Jenkins, funciona correctamente no fue necesario su uso. Con tener los fuentes respaldados en un SVN antes de realizar la formalización y en la herramienta oficial de la compañía nos ayudó a tener los fuentes centralizados y disponibles.

## PRÁCTICA #12: Juego de Planificación

Los Dueños de Producto nos envían solicitudes de cambio, los cuales son analizados por un Gestor de Mejoras. Cuando se termina la parte de análisis y se define una solución, el equipo Gestor de Mejoras asignado para esto escribe una Historia de Usuario. Es decir, en nuestro caso las historias de usuario no son escritas por nuestros clientes, sino que son escritas por el mismo equipo de desarrollo.

Aunque las Historias de Usuario deben escribirse en lenguaje de usuario, en nuestro caso la descripción se hace en un lenguaje lo más cercano al usuario, es decir, se hace en términos lo más comerciales que se pueda, y los criterios de aceptación se escriben en términos lo más técnico posible.

Las historias de usuario no son entregadas al Dueño del Producto, sino que son utilizadas como documentación de la solicitud y la solución a implementar. Al Dueño del Producto se entrega la estimación en jornadas y una descripción de la solución a implementar y qué contempla y no contempla la solución.

Hay oportunidades en que el Dueño del Producto solicita documentación específica. Ésta entonces es estimada y se realiza como parte del entregable a liberar.

Cuando el Dueño del Producto aprueba la implementación de la solución, se planifica definitivamente la fecha de inicio y término de la implementación y sus iteraciones, quiénes serán los participantes, es decir, quién codificará, quién certificará, quién escribirá los casos de pruebas, quién formalizará la solución, etc. Se presenta esta planificación en una reunión de equipo y se deja la planificación en el tablero Kanban.

Siempre es el Dueño del Producto quien ve las prioridades de las soluciones que se están implementando, lo que se negocia con ellos es cuántas soluciones pueden implementarse en simultáneo.

Tabla 46 Cualitativa de las Prácticas XP que se Utilizaron.

De las prácticas de XP, que no se utilizaron tenemos la siguiente revisión del cual hizo que la práctica no se utilizara:

Evaluación Cualitativa de las Prácticas XP
PRÁCTICA #1: Cliente en el Lugar
<p>Para la realidad de la compañía, que al tener un trato contractual con sus clientes no implica un compromiso de disponer de usuarios expertos del cliente que faciliten la construcción de las soluciones que ellos mismos solicitan.</p> <p>Además, nuestros clientes (a diferencia del que está ubicado en Chile), están en el extranjero por lo que es complicado estar en las dependencias del cliente sin que ellos tengan que pagar viáticos, hoteles, etc., producto del desplazamiento de las personas lo que encarecería el costo.</p> <p>Ahora, cuando nuestro cliente nos hace solicitudes de cambio con requerimientos poco definidos, entonces se negocia para continuar de todos modos, pero teniendo claro que en el camino se pueden necesitar aclaraciones de parte de los usuarios, o al estar construyendo puede cambiar el alcance o incluso el mismo Dueño del Producto puede querer cambiar el alcance, pero las fechas de entrega pueden cambiar.</p>
PRÁCTICA #2: Semana de 40 horas
<p>En la compañía, se ha llegado a tener una semana laboral de 42 horas. De todos modos, en el equipo de Soporte Mejoras, se busca ser flexibles con el horario laboral, para intentar ser compatible la vida personal con la laboral.</p>
PRÁCTICA #3: Metáfora
<p>El crear metáforas que ayuden a los usuarios a entender los cambios técnicos, no los utilizamos porque creemos que no será bien recibido por nuestros clientes, ya que cuesta que se asignen por parte de ellos personas, usuarios expertos, difícilmente consideraran esta práctica. Por lo tanto, el equipo decidió no utilizarla por ser poco practica para la realidad de la compañía.</p> <p>Pero sí tenemos la flexibilidad de hacer cuantas reuniones se puedan para que todos tengan claro los cambios que se entregarán.</p>

Tabla 47 Cualitativa de las Prácticas XP que se Utilizaron.

De Scrum, se utilizaron los siguientes elementos, con las siguientes evaluaciones:

### Iterativo e incremental

Cuando se planifica una implementación, se negocia con el Dueño del Producto las posibles iteraciones, donde siempre cada iteración implica software que tiene la capacidad de pasar a producción sin esperar las siguientes iteraciones.

En general este tema ha costado con los clientes, ya que ellos deben realizar las pruebas de aceptación y no siempre están disponibles para realizar estas pruebas por etapas ya que como en todas las empresas las personas son escasas y es complicado que sean asignadas por etapas. Pero siempre se intenta ser flexibles con los clientes y finalmente ellos tienen la decisión de cuántas iteraciones habrá para completar sus requerimientos.

### Equipo Multi – Funcional

Dependiendo de las habilidades de cada persona, la cultura que se formó en el equipo es que todos sus miembros deberían poder realizar la mayor cantidad de las actividades, de las cuales incluyen:

- ❖ Analizar requerimientos.
- ❖ Estimar.
- ❖ Codificar.
- ❖ Certificar.
- ❖ Gestionar a clientes.
- ❖ Gestionar los errores en las pruebas.
- ❖ Inspección de documentos y código.
- ❖ Documentar requerimientos, casos de pruebas, etc.
- ❖ Gestionar Riesgos.

Ahora siempre se aprovechan las mejores habilidades de cada uno, la intención es que una persona pueda ejecutar la mayor cantidad de estas actividades.

## De los Roles

En Scrum existen tres roles, los cuales se utilizaron de la siguiente forma:

Rol	Descripción
Scrum Master	<p>Este rol se tomó de forma parcial; en su reemplazo se definió el rol de Responsable de Equipo.</p> <p>Este rol es el responsable de supervisar las labores de cada miembro del equipo, así como también hacer las mismas labores del equipo, y velar por la mejora continua de los procesos y del trabajo, entre otras.</p>
Dueño de Producto	Para el equipo es quien entrega los requerimientos, y define la prioridad de los temas que revisa el equipo de Soporte Mejoras.
El Equipo	En el caso de Soporte Mejoras, el equipo de Scrum son los Gestores de Operadora, o sea quienes realizan todas las labores del equipo.



## Sobre las reuniones

Reuniones	Descripción
Reunión de Planificación del Sprint	<p>En este caso en el equipo sí se realiza esta reunión, pero no es con el Dueño del Producto.</p> <p>Una vez que el Dueño del Producto aprueba la implementación de una solución, el responsable realiza la planificación, se definen las tareas, fechas y los Gestores de Mejoras que serán responsables de cada una de las tareas. Esto nos ayudó a que cada persona tuviera claro cuál es su participación.</p>
Reunión Diaria	<p>Esta reunión es la que mejor evaluación tiene ya que permite que todos los miembros del equipo se enteren cómo van los otros miembros del equipo y den sus opiniones.</p>
Reunión de Retrospectiva	<p>Se realizan varias retrospectiva. Una de estas se realiza al termino de cada iteración y es para verificar las cosas buenas y malas que se realizaron en la iteración.</p> <p>Otra retrospectiva se realiza cada 6 meses en donde se revisa los motivos de los rechazos realizados por el Dueño del Producto, buscando prevenir estos en el futuro. Esto se hace mirando la Tasa de Rechazo del semestre.</p> <p>Además, se hace una retrospectiva al año para revisar el proceso y realizar las adaptaciones necesarias.</p> <p>Esta práctica tuvo una muy buena recepción por parte del equipo en donde cada uno podía hablar y dar sus opiniones.</p>

<b>Backlog del Producto</b>
En el equipo utilizamos un backlog para identificar todas las solicitudes que nos han llegado y que aún no hemos trabajado. Esto nos ayuda a tener a la vista y priorizados los temas que debemos seguir revisando cuando se dé la opción.
<b>Backlog del Sprint</b>
También tenemos este backlog. Cuando se planifica la implementación de una solución, se definen las tareas a realizar y estas son el backlog del sprint. En el mismo caso anterior esto nos ayuda a tener visualmente todas las tareas que están pendientes.  Los errores encontrados por los usuarios en la etapa de Pruebas de Aceptación, si son trabajados en la misma iteración forman parte de Backlog del Sprint, pero si estos cambios formaran parte de otra iteración éste es agregado al backlog del producto.
<b>Scrum taskboard</b>
En el equipo también tenemos un tablero Kanban que nos ayuda principalmente a ver la cantidad de tareas que tenemos en circulación. Esto nos ayuda mucho a tener visualmente la información y no estar dependiendo de una aplicación o planillas con esta información.
<b>Diagrama de Burndown</b>
Esta práctica sí se utilizó inicialmente, pero se desechó. Lo utilizábamos para medir el consumo de las tareas en el tiempo. Por ejemplo, si una implementación tomaba 16 tareas, esta se distribuía en el tiempo de acuerdo a lo planificado, es decir, el último día de planificación debería existir ninguna tarea pendiente. Ahora, en general esta actividad la dejamos de realizar debido a que la diferencia de lo planificado y lo que realmente ocurrió era mínima. En la mayoría de las veces nos sobraba tiempo y en pocas oportunidades nos pasábamos de lo planificado pero los márgenes siempre eran a juicio del equipo poca diferencia. Por este motivo dejamos de realizar esta actividad.

Tabla 48 Evaluación de las Prácticas de Scrum que se Utilizaron.

Del mismo Scrum, no se utilizaron las siguientes practicas:

Sobre las reuniones	
Reuniones	Descripción
Reunión de Planificación de la Entrega	<p>En Scrum, esta reunión es para que el Dueño del Producto entregue las prioridades al equipo. En nuestro caso no se hace una reunión, sino que se prefirió que a través de correos y conversaciones el Dueño del Producto informe las prioridades de las implementaciones.</p> <p>No hemos tenido problemas de coordinación por lo que trabajarlo de esta manera nos ha resultado positivo.</p>
Reunión de Revisión	<p>Esta reunión se realiza al interior del equipo, para revisar el entregable. A diferencia de Scrum esta no se hace con el Dueño del Producto, ya que a él más que revisar el contenido de lo enviado le interesa que esté bien por lo que las veces que lo intentamos no tuvimos buena recepción de ellos para hacerlo.</p> <p>Lo que sí se hace es explicar al equipo de Soporte Correctivo cuando la solución está a punto de pasar a producción, cuáles son los cambios realizados para que ellos puedan después mantener las aplicaciones.</p>

Tabla 49 Evaluación de las Prácticas de Scrum que no se Utilizaron.

# 5

## 5 Conclusión

---

El haber formalizado un proceso con prácticas ágiles adaptado a la realidad de TMAS, ayudó a ordenar el trabajo realizado independientemente de las personas que lo integran. En un inicio, y con el objetivo de entregar muy rápido soluciones, el equipo cayó en muchos errores que, aunque entregábamos las soluciones a tiempo, después estos nos eran rechazados, por lo que la calidad de los entregables no era buena. El proceso fue un extraordinario soporte en el que ayudó tener un marco de trabajo que orientó las actividades del equipo.

De la codificación de soluciones, de a pares no se usó se pospuso para usar en un tiempo ya que la compañía no estuvo de acuerdo en utilizar dos personas para implementar un requerimiento. Sin embargo, sí utilizamos una adaptación que es ir revisando el código, tratando de seguir la idea de la programación de a pares que es la colaboración entre los miembros del equipo. Por lo tanto, la práctica fue que se asigna a una persona que construya y a otros Gestores de Mejoras, para revisar el código y dar opiniones sobre cómo continuar la programación. Además, mientras dura la etapa de implementación del código fuente se va respaldando en un SVN, para que así cualquier integrante del equipo pueda ir revisándolo; con esto cumplíamos con la propiedad colectiva del código fuente.

En cuanto al diseño y la certificación, el Área de Desarrollo realizaba etapas separadas y documentadas de Análisis, Diseño y pruebas. Nosotros reunimos todo esto en un solo documento. Utilizando la idea de TDD, que primero se programan las pruebas y después su ejecución, con la intención que las pruebas guíen la construcción, se logró reunir en un solo documento el diseño de lo que se debe construir y los casos de prueba, para que los que van a construir también conozcan qué es lo que se va probar. La diferencia radicó en que el Área de Desarrollo utiliza una aplicación para documentar y para gestionar la cantidad de errores obtenidos en la etapa de construcción. Esto generaba una conducta de levantar en una incidencia muchos errores para no afectar los indicadores por lo cual el indicador nunca era real. Para el trabajo

realizado por nosotros no nos importaba registrar o medir los errores encontrados en la etapa de certificación, y en vez de utilizar una aplicación para registrar los casos de prueba se utilizaron documentos Word o Excel. Si hay incidencias, estas son conversadas con el equipo y no son registradas en aplicaciones. Lo que nos importaba era que el producto final fuera bueno y no controlar los errores en el proceso. Con esto logramos que ya no existiera una competencia perversa entre los que construyen y los que certifican y, además, como en ambas etapas las personas son del mismo equipo, tenían un objetivo común.

Inicialmente, cuando se exponían los análisis para lograr una estimación en cantidad de jornadas, la idea siempre fue realizar esta actividad a juicio experto, por lo que se consultaba a cada miembro del equipo cuánto estimaba para la implementación de la solución. Pero esto significaba que cuando hablaban los más expertos, los de menos experiencia repetían la estimación de los más expertos, por lo que no se lograba el objetivo de tener la opinión de todos los miembros del equipo. Para lograr este objetivo y hacer de la estimación algo más lúdico, se realizó una adaptación de Planning Póker donde en vez de cartas se utilizaron post-it, y en vez de tener la serie de Fibonacci, cada miembro escribe la cantidad de jornadas que estima él que utilizaría para implementar la solución propuesta. Con esto se logró tener la opinión de cada uno y, si la diferencia de jornadas era, a criterio del Responsable del Equipo, alta, se pide la opinión del por qué para él es esa cantidad de jornadas ya que puede ser que él consideró algo que los otros no han considerado, y se puede volver a jugar en caso de ser necesario.

Uno de los puntos que ayudó a lograr buenos resultados en la calidad de los entregables fue la motivación del equipo. El trabajar con personas motivadas es clave para lograr que las implementaciones salgan de buena manera y aportar valor al cliente. Este punto se trabajó escuchando y hablando con los miembros del equipo. Se usó la dinámica de mapas de afinidad (que se realiza en la etapa de la retrospectiva) se exponen qué es lo que ellos requieren para poder hacer mejor su trabajo. Después se realizan acciones para poder lograr alguna de estos temas, como, por ejemplo, flexibilidad laboral, trabajo por objetivos, etc.

El incorporar prácticas al equipo de Soporte Mejoras, ha sido de un alto beneficio para el trabajo que se realiza, ya que ha ayudado a ordenar, mejorar la eficiencia, sistematizar, innovar y aprovechar las mejores habilidades de cada persona, ha ido de manera sistemática mejorando también la calidad de los entregables y la calidad del servicio entregado a los clientes. Se dio a la

compañía una oportunidad se poder dar estimaciones más acordes a ciertos requerimientos de los clientes, esto permitió que estos sean aceptados e implementados.

Los siguientes pasos que vienen es lograr mejorar los indicadores de calidad que, aunque no se pretende que estos sean cero, pero que sí tiendan a cero ya que siempre se está trabajando con requerimientos inestables y poco claros, pero como son de menor tamaño el riesgo también es menor. En la compañía se ha dado que los clientes están cada vez solicitando requerimientos más pequeños, y según lo indicado por ellos esta será la tónica de aquí hacia el futuro. La compañía ha visto la necesidad de cambiar la forma de hacer software y ser más flexibles, debido a que el trabajo y los frutos realizados por el equipo de Soporte Mejoras, están tomando más relevancia. El siguiente paso es que todos los equipos de la compañía que desarrollan software lo hagan con prácticas ágiles que, a opinión de los Gerentes de Cuentas, han dado al cliente mucha flexibilidad por los temas que nosotros hemos revisado, por lo que se viene ahora para mí es liderar que todos los equipos de la compañía incorporen Prácticas Ágiles en su trabajo.



## Bibliografía

---

- [1] Scrum & Extreme Programming, para Programadores, Eugenia Behit, 2012
- [2] Scrum Manages, Gestión de Proyectos, Juan Palacio, Claudia Ruata, 2011.
- [3] En Busca de la Excelencia del Código, Raúl Hernanz Serrano, 2012.
- [4] eXtreme Programming, José Carlos Cotizo Pérez, Expósito Gil, Miguel Ruiz Leyva.
- [5] Definición de PPM. <http://www.tcpsi.com/vermas/hp-ppm.htm>.
- [6] Tesis: Diseño de una Metodología Ágil de Desarrollo de Software, Schenone Marcelo Hernán. Universidad de Buenos Aires. 2004.
- [7] Ingeniería de Software, Alejandra Aparicio, 2012.

### Artículos:

- [8] Introduciendo Conceptos de Metrología en el Diseño de Medidas de Software, Nelly Condori-Fernández, Oscar Pastor, Alain Abran, Asma Sellami.
- [9] Modelo para Almacenar y Recuperar Métricas de Software, Dr. Enrique Luna Ramirez, Dr. Francisco J. Alvarez, Dr. J. Mauricio Espinoza Mejía, Humberto Ambriz, J. Antonio Nungaray. 2010.
- [10] Adopción de Metodologías Ágiles de Desarrollo, Carlos Nuño, Héctor Fragoso. 2014.
- [11] Calidad de software e Ingeniería de Usabilidad, Mascheroni, M.; Greiner, C.; Petris, R.; Dapozo, G.; Estayno M. 2012.
- [12] Un Marco Conceptual para la Definición y Explotación de Métricas de Calidad, L. A. Olsina, M. F. Bertoa, G. J. Lafuente, M. A. Martín, M. Katrib, A. Vallecillo.
- [13] SPEM: Software Process Engineering Metamodel, Víctor Hugo Menéndez Domínguez, María Enriqueta Castellanos Bolaños, Facultad de Matemáticas, Universidad Autónoma de Yucatán, Mérida, México.

- [14] Metodologías Ágiles en el Desarrollo de Software, José H. Canós, Patricio Letelier, M<sup>a</sup> Carmen Penadés. DSIC -Universidad Politécnica de Valencia.
- [15] Metodologías Tradicionales v/s Metodologías Ágiles, Roberth G. Figueroa, Camilo J. Solís, Armando A. Cabrera. Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación.
- [16] Metodologías Ágiles, Amaro Calderón, Sarah Dámaris, Valverde Rebaza. Jorge Carlos. Universidad Nacional de Trujillo, Facultad de Ciencias Físicas y Matemáticas, Escuela de Informática. Perú, 2007.
- [17] Modelos y Métricas para Evaluar Calidad de Software. Marcelo Estayno; Gladys Dapozo; Liliana Cuenca Pletch; Cristina Greiner.





### A. PPM de HP

PPM: Project and Portfolio Management, es una solución de HP, para Gestión de Proyectos y Servicios. Permite la consolidación de la visibilidad y los datos dentro de la cartera empresarial.

PPM gestiona procesos de acuerdo a workflows y reglas personalizables, permitiendo el seguimiento de prácticas de Gobierno TI (IT Governance), aplicadas a los distintos componentes que gestiona un departamento TI.

Con PPM, se gestionan las peticiones que llegan a la compañía (Gestión de la Demanda). Permite controlar el uso de recursos humanos y financieros de una organización.

Como PPM, se puede gestionar incidencias a través de un workflow, se pueden crear ticket de distintos tipos, en la compañía se han generado los siguientes:

- ❖ Incidencias: para que nuestros clientes levanten incidencias de SCL.
- ❖ Consultas: para que la operadora pueda realizar preguntas sobre las funcionalidades de SCL.
- ❖ Problemas: para los envíos que se realicen como Gestión Problemas.
- ❖ Mejoras: este tipo de ticket no es visible por los clientes de la compañía, y es para el uso del equipo de Soporte Mejoras, para documentación de los archivos al interior del equipo.



## B. Tablero Visual del Equipo de Mejoras

Tablero Visual, que sirve como herramienta de gestión de asignación y carga del equipo. Un tablero es un método para gestionar el trabajo, con énfasis en la entrega justo a tiempo, mientras no se sobrecargan a los miembros del equipo. En este enfoque, el proceso, desde la definición de una tarea hasta su entrega al cliente, se muestra para que los participantes lo vean y los miembros del equipo tomen el trabajo de una cola.

El Tablero del equipo tiene las siguientes secciones:

- ❖ Backlog: solicitudes pendientes sin revisar.
- ❖ Pendientes: Temas ya trabajados que están esperando a ser atendidos (como construcciones), por ejemplo, temas ya analizados que han sido entregados al Dueño del Producto, pero él no ha dado el VoBo, para su implementación.
- ❖ Trabajando: en esta sección esta los nombres de cada persona miembro del equipo y tiene a su derecha los temas asignados a cada uno.
- ❖ En Revisión: tareas terminadas que deben ser revisadas o inspeccionadas.
- ❖ Terminadas: tareas que sean sido terminadas, pero no han sido enviadas al Dueño del Producto.
- ❖ Entregadas: paquetes entregados al Dueño del Producto.
- ❖ Planificación: planificaciones de las implementaciones a realizar.
- ❖ Gestión de Riesgos: zona en donde se escriben en post-it los posibles riesgos y sus mitigaciones.
- ❖ Paso a Producción: implementaciones que pasaron a producción, y se tiene o se está trabajando en la actualización de ambientes y repositorio de fuentes al interior del equipo.
- ❖ Cancelación: implementaciones que deben ser canceladas porque el Dueño del Producto, no las paso a producción.

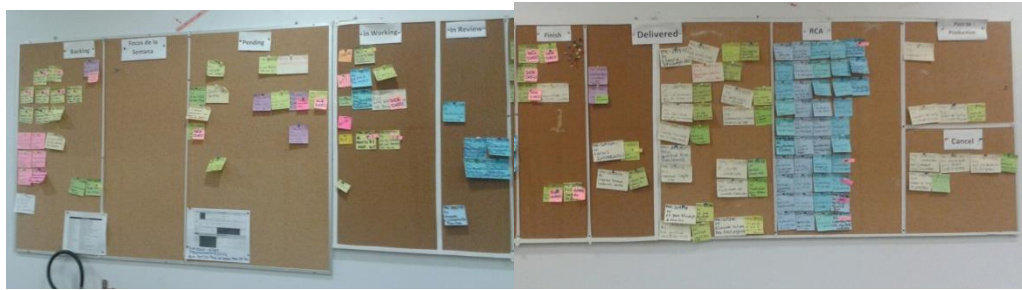


Figura 13. Kanban del Equipo.



## C. Plantilla de Planificación

Es un documento que sirve para poder conocer la carga del equipo en el tiempo, como también tener registro de las fechas de inicio y término de las implementaciones de las solicitudes que se implementara una solución.

La plantilla es un calendario, en el cual se puede revisar lo que está comprometido. Por ejemplo, si hay solicitudes ya planificadas éstas deben estar en este archivo para así se tenga registro de lo que se está trabajando como también tener identificados cada uno de los Gestores de Mejoras involucrados. Cuando se planifica una solicitud hay que llevar todas las tareas a realizar a un nivel de día, es decir, la planificación dirá día a día lo que se tiene que hacer.

Día	Enero-2016							
	01	02	03	04	05	06	07	08
German								
Gabriel	LUA	LUA	LUA	LUA	LUA			
Miguel	OOSS-PROV-CONST	OOSS-PROV-CONST	OOSS-PROV-CONST	OOSS-PROV-CONST	OOSS-PROV-FAU#1			
Anita		SS-CP#2	SS-CP#2		Bajas-no-asumidas			
Gonzalo-C	SS-AMB#2			SS-CONST#2	SS-FAU#2			
Gonzalo-S	Encolamiento	Gestor-Tipo-9	ERROR-3018					
	11	12	13	14	15	16	17	18
German				SS-INSPE#2				
Gabriel	LUA	LUA	LUA	LUA	LUA			
Miguel	OOSS-PROV-DOC	OOSS-PROV-DOC	OOSS-PROV-DOC	Bajas-No-Asum.	Bajas-No-Asum.			
Anita	Hibernación	SS-CERT#2	SS-CERT#2					
Gonzalo-C	SS-CONST#2	OOSS-PROV-CERT	OOSS-PROV-CERT	OOSS-PROV-CERT	SS-FAU#2			
Gonzalo-S	Encolamiento	Encolamiento	Encolamiento	SitABP	SitABP			
	18	19	20	21	22	23	24	25
German	SS-INSPE#2			OOSS-PROV-				

Figura 14. Plantilla de Planificación.

# D

## D. Plantilla de Historias de Usuario

Las historias de usuario son una forma rápida de administrar los requisitos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos.

La plantilla usada por el equipo de Soporte Mejoras es la siguiente (junto con la descripción de cada uno de los ítems):

Enunciado de la Solicitud	
Nombre de la solicitud	Cuál es el nombre de la solicitud, debe ser un nombre lo más corto posible pero que sea representativa de la solicitud.
Descripción	Esta es la descripción de lo que se va a realizar, este debe ser en lenguaje del usuario. Debe alejarse lo más que se pueda de aspectos técnicos.
Criterio Aceptación	Qué es lo que se debe probar. Qué es lo que el usuario va a probar y qué es lo que espera como resultado. Esto puede ser en términos técnicos, para que quien construya tenga antecedentes de lo que se debe implementar
Criterio Aceptación	
Objetivos	Para qué se está haciendo el cambio. Esto también puede ser expresado de manera técnica.

Tabla 50 Plantilla de HU.

El nombre del archivo debe seguir el siguiente formato

<código de la operadora>-<número del ticket de mejoras de PPM>\_HU.

Ejemplo: TMC-20856\_HU.



## E. Estimación de la Solución

Se determinan la cantidad de jornadas de la construcción, complejidad y si dado el riesgo de la implementación ver si es factible, implementarla. No se tiene ningún artefacto, ya que el responsable del equipo obtiene estos valores de la reunión.

### *Estimación Construcción*

Técnica de estimación utilizada por el equipo, la cual está basada en Planning Póker.

Se reúne al equipo, y una vez que se explica la solución a implementar, se sigue la siguiente dinámica:

- i. Se entrega a cada persona que participa en el equipo un post-it y un plumón (debe ser plumón para que lo que escriba sea visto por todos).
- ii. Cada uno escribe sin que otro lo vea cuántas jornadas según su criterio tomará realizar la implementación de la solución (es solo la construcción). Es un número entero ejemplo 1, 2 o 7, etc., no puede ser 1,5. Si alguien no tiene los conocimientos técnicos para construir la solución debe escribir en el post-it un signo de interrogación.
- iii. Cuando todos terminan de escribir se deben dar vuelta los papeles (como cartas de póker) para que todos lo vean.
- iv. Si a juicio del Responsable del Equipo, si la diferencias en los valores no es muy grande se puede sacar un promedio y este sería el costo de la construcción, si hay mucha diferencia el Responsable del Equipo debe preguntar los motivos para que se dé esta estimación, tanto la mayor como la menor, si hay argumentos que los miembros del equipo no consideraron, y si es necesario repetir la votación. Hasta obtener el valor de la construcción.

## ***Complejidad de la Solución***

La complejidad, tiene relación a qué tan difícil para el equipo es la implementación de la solución. Por ejemplo, una implementación puede tener muchas jornadas de construcción, pero como se domina la tecnología y la funcionalidad puede ser de baja complejidad.

- ❖ S: baja complejidad.
- ❖ M: mediana a baja complejidad.
- ❖ L: mediana a alta complejidad.
- ❖ XL: alta complejidad.





## F. Plantilla de Estimación Final

Este documento registra la estimación total a utilizar para implementar la solución de la solicitud. Se debe sumar la cantidad de jornadas de cada uno de los ítems y con esto se tendrá la cantidad de jornadas totales de la solución.

Tabla en donde se ingresan la cantidad de Jornadas total a consumir para implementar la solución de la solicitud.

Para obtener la estimación final hay que asignar jornadas a cada uno de estos ítems:

- ❖ **Análisis:** cantidad de jornadas que se utilizan para analizar el requerimiento más el tiempo de todas las personas que intervinieron en este análisis.
- ❖ **Ambiente:** depende del tipo de lenguaje es cuánto costará en armar el ambiente. Aquí algún miembro del equipo que tenga experiencia en ambientar en la tecnología que se implementará la solución deberá asignar la cantidad de jornadas.
- ❖ **CP (Casos de Prueba):** 30% de la construcción. Se trata de escribir casos de prueba técnicos que orienten al desarrollador a la implementación de la solución.
- ❖ **Construcción:** esta cantidad de jornadas sale del Planning Póker.
- ❖ **Certificación:** 30% de la construcción. Es la cantidad de jornadas que se tendrá para ejecutar y documentar los casos de prueba.
- ❖ **Documentación:** 10% de la construcción. Es para describir la solución a implementar. Esta documentación es la que se enviará al cliente.
- ❖ **Inspección:** 20% de la construcción. Es para inspeccionar el código, la documentación, los casos de prueba y la formalización.
- ❖ **Formalización:** Este tiempo dependerá de la cantidad de piezas que se estiman crear o modificar. Con este tiempo se debe crear los paquetes instaladores y de reversa. Versionar el cambio en la herramienta de versionamiento, más el tiempo de auditoria del paquete a enviar al cliente.

- ❖ Holgura: 40% de la construcción. Es el tiempo que se tiene como respaldo por actividades no contempladas en la planificación.

Se suman la cantidad de jornadas de estos ítems y este es el tiempo total en jornadas a utilizar para implementar la solución.



## G. Plantilla de Casos de Prueba

La definición de las pruebas antes de la construcción tiene como objetivo el poder desarrollar para cumplir una prueba, es decir, al escribir las pruebas se está describiendo cómo la persona que construya el software lo haga siguiendo los lineamientos de los casos de prueba. Por este motivo los casos de prueba deben ser lo más técnico posible.

Este documento tiene el siguiente formato:

P00<correlativo de la prueba debe partir por la prueba 1>.- Nombre de la prueba.

### **Descripción**

<Descripción de la prueba>

### **Casuística**

<Que casuísticas deben cumplir las pruebas>

### **Resultado Esperado**

<Resultado Esperado General de la prueba>

<b>Nombre paso</b>	<b>Descripción</b>	<b>Resultado esperado</b>
<b>PASO</b> <b>&lt;correlativo del paso&gt;</b>	<descripción del paso>	<resultado esperado del paso>

### **Evidencia de las pruebas:**

<Se debe evidenciar con imágenes o videos la ejecución de la prueba que demuestre su correcta ejecución>.



## H. Plantilla de Documento ADP

ADP: significa Análisis, Diseño y Pruebas, es un documento definido por la compañía, para describir información relevante a la implementación pero que tiene como destinatario al cliente. Por lo que el lenguaje usado en él debe ser en lenguaje del Cliente.

Debe contener la solución funcional y técnica, más las evidencias de pruebas del documento de casos de prueba.



## I. Proceso del Área de Desarrollo

El flujo de trabajo del proceso tradicional que tiene TMAS, se divide en 4 etapas, las cuales se muestran en la siguiente ilustración.



Figura 15. Flujo de Trabajo de TMAS.

El proceso comienza con las operadoras enviando a TMAS requerimientos, los cuales son analizados por el Área de Clientes (por un Consultor), los cuales a su vez se transforman en los Requerimientos de la Operadora (RM<sup>11</sup>), y de estos en Requerimientos de Sistema (RMS<sup>12</sup>). Allí se expresan los requerimientos funcionales, no funcionales y los casos de uso. Con los casos de uso se estiman los requerimientos con puntos de función, y se entrega a la operadora la estimación técnica y económica presupuestaria (ETEP). En la siguiente ilustración, se muestran los pasos de esta etapa.

---

<sup>11</sup> RM: Requerimientos Maestros de Usuario.

<sup>12</sup> RMS: Requerimientos Maestros de Sistemas.

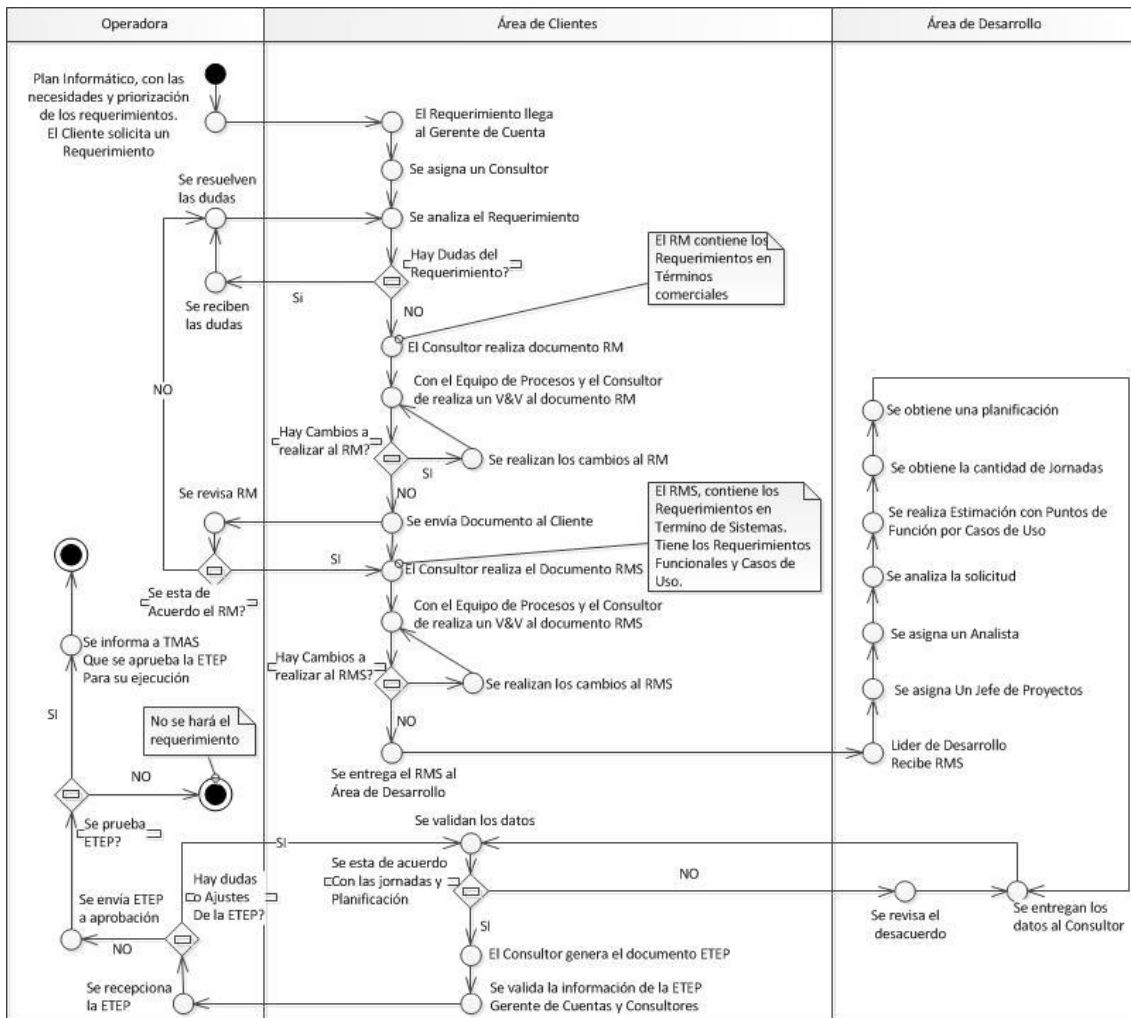


Figura 16. Etapa de Gestión Previa.

La ETEP, funciona como un presupuesto que la operadora puede aprobar o no. De aprobarse los requerimientos, éstos son implementados por el Área de Desarrollo, los cuales generan documentos de análisis y diseño, manuales, pautas de instalación y además certifican las aplicaciones. Este flujo de trabajo se basa en RUP<sup>13</sup>. Este proceso se describe a grandes rasgos en la siguiente ilustración.

<sup>13</sup> RUP: Rational Unified Process, Best Practices for Software Development Teams, Rational Software White Paper, 1998, IBM, Pag. 1.

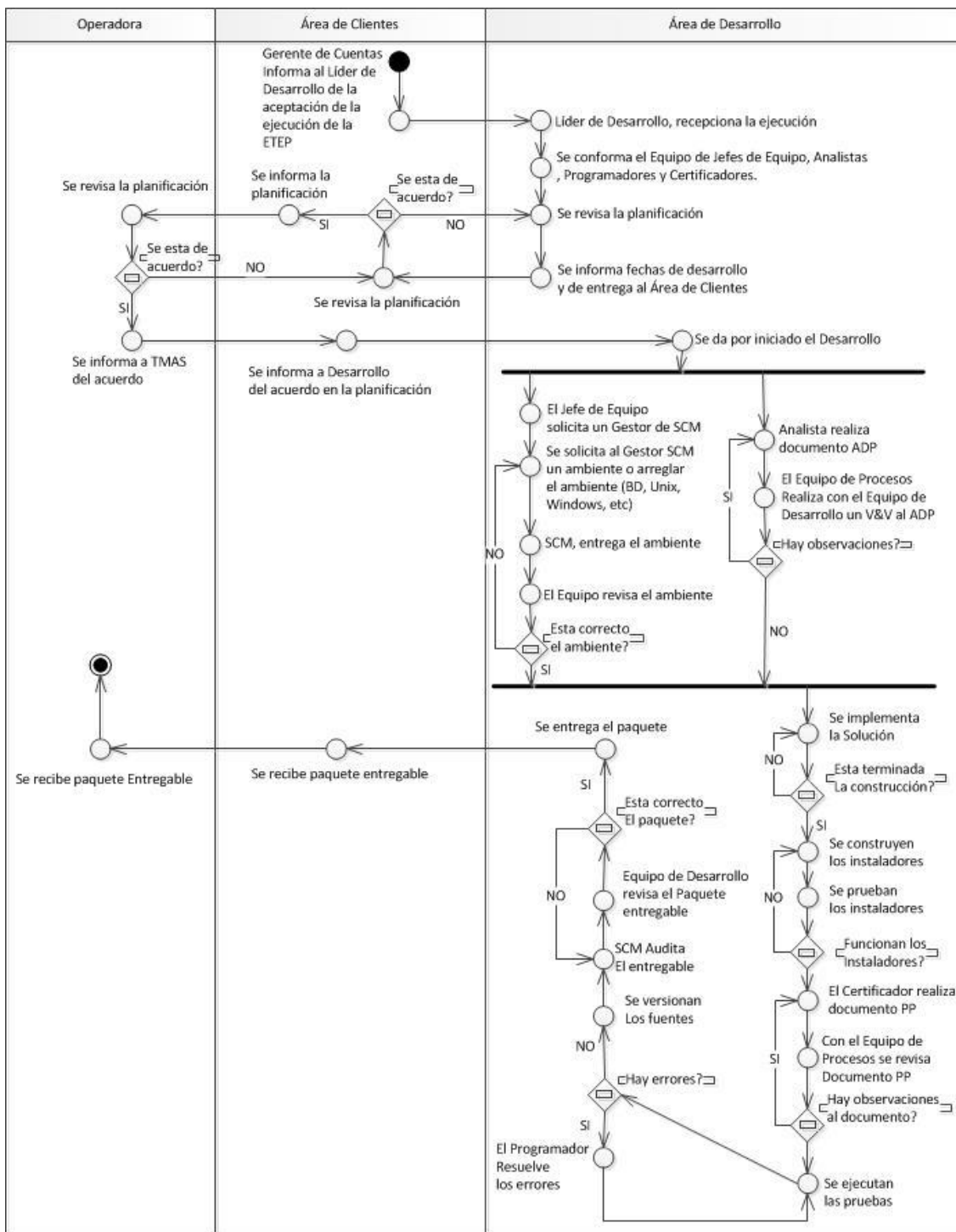


Figura 17. Etapa de Desarrollo.

Una vez terminado el desarrollo, el requerimiento pasa a implantación que pertenece al Área de Clientes. En esta etapa de implantación la Operadora realiza las pruebas de aceptación del sistema. Por lo que dentro de esta etapa





Cada una de estas etapas además de los pasos mencionados, también se debe ir actualizando la herramienta PPM, y esto es revisado por la PMO, solicitando aclaraciones en las distintas reuniones de seguimientos.