

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



Facultad de Ingeniería en Electricidad y Computación

**“MODELAMIENTO BPM PARA LA
MEJORA DEL SUBPROCESO DE PRUEBAS DEL PROCESO
DE DESARROLLO DE SOFTWARE DENTRO
DE UNA EMPRESA DE SOLUCIONES
INFORMÁTICAS USANDO LA METODOLOGÍA DE
AGILE TESTING”**

TRABAJO DE TITULACIÓN

PREVIO A LA OBTENCIÓN DEL TÍTULO DE

MAGÍSTER EN SISTEMAS DE INFORMACIÓN GERENCIAL

AUTORES

GURUMENDI PATIÑO JEANNETTE ALEXANDRA

BARBA GARRIDO PABLO JAVIER

Guayaquil – Ecuador

2021

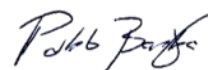
AGRADECIMIENTOS

Agradezco en primer lugar a Dios, por brindarme la oportunidad de culminar este reto. A mi familia por las horas robadas de estar junto a ellos y por el amor que me demostraron ayudándome para poder cumplir esta meta.



LSI. Jeannette Gurumendi Patiño

Agradezco a Dios ya que sin su voluntad nada es posible. A los profesores de la maestría que supieron impartir su conocimiento. A mi familia, que son siempre mi apoyo.



Ing. Pablo Barba Garrido

DEDICATORIAS

Dedico este trabajo a Dios por darme todas las herramientas para cumplir esta nueva meta.

A mis padres por ser la guía en mi vida animándome siempre a seguirme preparando y no desfallecer para cumplir mis metas. Mi padre desde el cielo guiándome y mi madre dándome siempre sus sabios consejos.

A mi esposo por apoyarme en todo momento para que pueda realizar la maestría.

A mis hijos Gabriela, Matthias y Horacio porque son las fuerzas que me impulsan a realizar nuevos retos.

LSI. Jeannette Gurumendi Patiño

DEDICATORIAS

Dedico este trabajo a mi esposa Jessenia y mi hijo Pablo, quienes tuvieron que esperar por mí en tantas ocasiones, pero han tenido la fuerza para darme su apoyo en esta jornada.

Ing. Pablo Barba

TRIBUNAL DE SUSTENTACIÓN



Mgtr. Lenin Freire Cobos

COORDINADOR MSIG



Mgtr. Juan Carlos García Plúa

DIRECTOR DEL TRABAJO DE TITULACIÓN



Mgtr. Robert Andrade Troya

MIEMBRO DEL TRIBUNAL

RESUMEN

El presente trabajo se desarrolla con la intención de poder realizar un análisis del proceso principal del negocio de la empresa como lo es el proceso de desarrollo de software y basarnos en uno de sus subprocesos más críticos como lo es el subproceso de pruebas y poder definir mejoras que le sirvan al equipo para ser más productivos y poder generar un producto de mejor calidad.

Este proyecto involucra al personal de los diversos equipos que interactúan dentro del subproceso de pruebas y las actividades que estos realizan son diagramadas mediante un modelamiento BPM que permite tener una buena idea de cuál es la situación actual.

Haciendo un análisis de los problemas críticos que se presentan en el subproceso actual y considerando las mejores prácticas utilizadas en Agile Testing de acuerdo con las necesidades del equipo, se rediseña el subproceso de pruebas tomando en cuenta que la empresa ya trabaja con bases de Scrum pero aún debe madurar mucho más en el uso de este marco de trabajo.

El subproceso rediseñado pasa a una etapa de implementación en la cual se lo incluye para ser utilizado en el proceso regular de desarrollo de software de la empresa de donde se obtienen datos para las diferentes métricas que son analizadas y comparadas con la actividad histórica de la compañía.

ÍNDICE GENERAL

RESUMEN	V
ÍNDICE GENERAL	VI
ÍNDICE DE FIGURAS	X
ÍNDICE DE TABLAS	XII
INTRODUCCIÓN	XV
CAPÍTULO 1	1
1 GENERALIDADES.....	1
1.1 ANTECEDENTES.....	1
1.2 DESCRIPCIÓN DEL PROBLEMA	3
1.3 SOLUCIÓN DEL PROBLEMA.....	4
1.4 OBJETIVOS ESPECÍFICOS.....	6
1.5 METODOLOGÍA	6
CAPÍTULO 2	9
2 MARCO TEÓRICO	9
2.1 METODOLOGÍA DE GESTIÓN DE PROCESOS (BPM)	9

2.2	MARCO DE TRABAJO SCRUM	11
2.3	PROCESO SCRUM	13
2.4	AGILE TESTING	16
CAPÍTULO 3		24
3	ANÁLISIS Y LEVANTAMIENTO DEL PROCESO ACTUAL	24
3.1	MODELO MACRO DEL PROCESO DE DESARROLLO	24
3.2	MODELO AS-IS SUBPROCESO DE PRUEBAS	26
3.3	ANÁLISIS DEL MODELO	28
CAPÍTULO 4		35
4	ANÁLISIS DE MEJORES PRÁCTICAS DE AGILE TESTING	35
4.1	ANÁLISIS DE ESTRATEGIAS DE AGILE TESTING	35
4.2	ACTIVIDADES DE AGILE TESTING EN LA REUNIÓN DE PLANIFICACIÓN DEL RELEASE	52
4.3	ACTIVIDADES DE AGILE TESTING DURANTE LA PLANIFICACIÓN DEL SPRINT	54
4.4	ACTIVIDADES DE AGILE TESTING DENTRO DEL SPRINT	57
4.5	ACTIVIDADES DE AGILE TESTING EN LA LIBERACIÓN.....	63

4.6	PRÁCTICAS AGILE TESTING COMPRANDO EL PROCESO ACTUAL ...	65
4.7	PRÁCTICAS DE AGILE TESTING PROPUESTOS PARA SOLUCIONAR PROBLEMAS DEL PROCESO ACTUAL	69
	CAPÍTULO 5	74
5	PROPUESTA DE MEJORA DEL PROCESO.....	74
5.1	REDISEÑO DEL PROCESO.....	74
5.2	IMPACTO DEL REDISEÑO	79
5.3	PROCESO MACRO DEL PROCESO DE DESARROLLO MEJORADO ...	82
5.4	MODELO TO - BE DEL SUBPROCESO DE PRUEBAS.....	84
	CAPÍTULO 6	87
6	IMPLEMENTACIÓN Y RESULTADOS	87
6.1	DEFINICIÓN DE LA IMPLEMENTACION	87
6.2	ANÁLISIS DE RESULTADOS.....	92
6.3	COMPARACIÓN DE MÉTRICAS.....	99
	CONCLUSIONES Y RECOMENDACIONES.....	106

ABREVIATURAS Y SIMBOLOGÍAS

AS-IS	As Is - Proceso "Como esta actualmente"
ATDD	Acceptance Test Driven Development - Desarrollo orientado a pruebas de Aceptación
BPM	Businness Process Management - Gestión de proceso de Negocios
PO	Product Owner - Propietario del Producto
RAM	Random Access Memory - Memoria de acceso Aleatorio
ROM	Read Only Memory - Memoria de solo Lectura
TDD	Test Data Driven - Desarrollo guiado por pruebas de Software
TFS	Team Foundation Server
TO-BE	To Be - Proceso "Como se espera que sea"
UAT	User Acceptance Testing - Pruebas de Aceptación de Usuario
UI	User Interface - Interfaz de Usuario

ÍNDICE DE FIGURAS

Figura 1.1 Autoevaluación Agile	3
Figura 2.1 Ciclo de Vida BPM	10
Figura 2.2 Proceso Scrum.....	14
Figura 2.3 Pruebas en método Cascada	18
Figura 2.4 Pruebas en metodologías ágiles	19
Figura 2.5 Interacción de roles	19
Figura 2.6 Proceso de pruebas ágiles	21
Figura 3.1 Equipos proceso de desarrollo	25
Figura 3.2 Proceso desarrollo de software	25
Figura 3.3 Resultado de encuesta.....	31
Figura 3.4 Respuestas afirmativa etapa planificación.	31
Figura 3.5 Respuestas afirmativas etapa implementación.	32
Figura 3.6 Respuestas afirmativas etapa de cierre.	32
Figura 4.1 Cuadrantes de Agile Testing	39
Figura 4.2 Cuadrantes con cara al apoyo al desarrollo	40

Figura 4.3 Cuadrantes con cara que critican el producto	43
Figura 4.4 Estrategias de pruebas ágiles	46
Figura 4.5 Pirámide de automatización de pruebas	49
Figura 4.6 Principales actividades del Agile Tester	50
Figura 6.1 Comparación de Evaluación de la agilidad.....	100
Figura 6.2 Comparación agilidad.....	101
Figura 6.3 Comparación errores en producción	102
Figura 6.4 Comparación puntos por sprint	103
Figura 6.5 Métrica Origen de errores dentro del sprint	104
Figura 6.6 Métrica de Automatización de pruebas unitarias	105

ÍNDICE DE TABLAS

Tabla 1 Testing Manifiesto	17
Tabla 2 Color referencia de Etapas	26
Tabla 3 Resultado de encuesta	30
Tabla 4 Comparación Tester y Agile Tester	36
Tabla 5 Comparación Desarrollador y	37
Tabla 6 Comparación entre PO	38
Tabla 7 Tipos de pruebas cuadrante 1	41
Tabla 8 Tipos de pruebas Cuadrante 2	42
Tabla 9 Tipos de pruebas del Cuadrante 3	43
Tabla 10 Tipos de pruebas del cuadrante 4	44
Tabla 11 Estrategias por Etapas	47
Tabla 12 Relación pirámide de automatización con el equipo	49
Tabla 13 Proceso Agile Testing macro	51
Tabla 14 Actividades dentro de la reunión del release	54
Tabla 15 Actividades en la planificación del sprint	57
Tabla 16 Actividades dentro del sprint.....	62

Tabla 17 Actividades de liberación del sprint.....	65
Tabla 18 Comparación Cuadrantes con Proceso pruebas actual.....	65
Tabla 19 Comparación Estrategias del proceso	66
Tabla 20 Comparación Actividades de planificación	67
Tabla 21 Comparación actividades dentro del Sprint	68
Tabla 22 Comparación actividades liberación con proceso actual	68
Tabla 23 Practicas de Agile Testing a problemas.....	70
Tabla 24 Practicas de Agile Testing a problemas críticos de implementación.....	71
Tabla 25 Practicas de Agile Testing a problemas críticos en revisión	72
Tabla 26 Objetivos proceso.....	74
Tabla 27 Roles y actores.....	75
Tabla 28 Relación Actividad con Agile Testing	76
Tabla 29 Actividades durante la planificación.....	77
Tabla 30 Actividades durante la implementación	78
Tabla 31 Actividades durante la Revisión del sprint	79
Tabla 32 Impacto en la planificación	80
Tabla 33 Impacto durante implementación.....	81

Tabla 34 Impacto finalización	81
Tabla 35 Roles proceso mejorado.....	82
Tabla 36 Agilidad por área Sprint 1	93
Tabla 37 Errores dentro del sprint 1	94
Tabla 38 Agilidad Sprint 2	95
Tabla 39 Errores dentro sprint 2.....	97

INTRODUCCIÓN

En la actualidad las empresas ecuatorianas que se dedican al desarrollo de software se enfrentan a varios desafíos que en muchas ocasiones tienen que ver con la evolución de sus equipos de trabajo y una constante búsqueda de poder hacer sus procesos más eficientes para poder responder a las altas exigencias del mercado.

Una de las tendencias más populares y que ha tenido una gran aceptación a nivel mundial es la adopción de metodologías ágiles por su inherente capacidad de adaptar la forma en que se trabaja según las condiciones de los proyectos, proporcionando grandes beneficios a través de esta flexibilidad.

El presente proyecto nace de la necesidad de una empresa nacional dedicada al desarrollo de software de ofrecerle a su equipo de trabajo una forma de poder incrementar su madurez respecto a la aplicación de una de estas metodologías ágiles, hablando específicamente de Scrum, que es el marco de trabajo con el que vienen desarrollando su producto desde hace dos años y de esta manera conseguir mejorar los puntos resultantes más críticos. En este proceso la empresa se encuentra con que sus equipos no están conectados de la mejor manera y particularmente el equipo de pruebas sufre por la falta de estructura de las actividades que realizan en la actualidad y es en esta parte donde surge la propuesta de rediseñar el subproceso de pruebas que es crítico para el proceso general de desarrollo y la mejora de sus resultados.

Para presentar el desarrollo de este proyecto, hemos dividido el presente documento en 6 capítulos definidos de la siguiente forma:

- Capítulo 1: Incluye los antecedentes, situación actual y descripción del problema.
- Capítulo 2: Incluye todo lo relacionado al marco teórico revisado en el desarrollo de este proyecto.
- Capítulo 3: Incluye el análisis del proceso tal como se lo encontró, y el modelado del mismo utilizando BPM para su mejor comprensión.
- Capítulo 4: Incluye el análisis de las mejores prácticas de Agile Testing que serían aplicables a la mejora del subproceso.
- Capítulo 5: Incluye la explicación de el rediseño del proceso y el modelado del nuevo proceso utilizando BPM para su mejor comprensión.
- Capítulo 6: Incluye el análisis de los resultados, comparativa de métricas y conclusiones.

CAPÍTULO 1

1 GENERALIDADES

1.1 ANTECEDENTES

El presente trabajo se lleva a cabo dentro de una empresa ecuatoriana que tiene más de 20 años activa en la industria del desarrollo de software para uso empresarial.

Para poder entender mejor la situación actual de la empresa y su proceso de desarrollo es necesario describir la estructura de las áreas y la forma en que se trabaja. La empresa cuenta con alrededor de 50 empleados divididos en equipos que son el equipo de desarrollo de software, pruebas, servicios y consultoría, ventas y personal administrativo.

De los equipos mencionados anteriormente, enfocaremos nuestro proyecto en el trabajo ejercido por dos de ellos, el equipo de desarrollo de software que es

en donde se efectúan directamente las tareas de programación y el equipo de pruebas que revisa el producto entregado por los desarrolladores y hace las pruebas necesarias antes del paso final a producción. Los dos equipos se encuentran a cargo de un director de desarrollo que coordina las actividades de los grupos.

La planificación del desarrollo se realiza mediante reuniones con frecuencia mensual en las que se define el plan de trabajo del período venidero. El plan de trabajo incluye cuáles serían los puntos más importantes en los que los equipos deben trabajar, sean estos, nuevas funcionalidades para las aplicaciones, corrección de errores conocidos, mejoras en las interfaces o experiencia de usuario o incluso el desarrollo de una aplicación nueva. En el proceso de definir el desarrollo que se va a realizar, se determina cuáles son los objetivos inmediatos en detalle, el alcance de cada uno de estos y los recursos, tanto del área de desarrollo como de calidad, que se asignan para trabajar en cada parte del proyecto. Finalmente se establecen los tiempos de entrega de cada punto tomando en cuenta el desarrollo, la ejecución de pruebas unitarias y el trabajo de revisión y pruebas finales que realiza el área de pruebas de forma independiente.

Desde aproximadamente dos años se ha empezado a adoptar prácticas de Scrum como marco de trabajo para el desarrollo de los productos. Aunque el equipo ya cuenta con algo de experiencia, la empresa sigue trabajando para buscar aumentar la madurez del mismo respecto a este marco de trabajo y

poder mejorar tanto la rapidez de desarrollo como la calidad del producto que entrega a sus clientes.

1.2 DESCRIPCIÓN DEL PROBLEMA

Es común ver como hay muchas empresas que han empezado a utilizar metodologías ágiles en sus proyectos de desarrollo de software. Entre estas, la Empresa XYZ, ha adoptado la metodología SCRUM, pero en su proceso se ha encontrado con el incremento del número de reclamos de los clientes por la calidad del software entregado.

El área de Desarrollo de Software se encuentra conformado por el equipo de desarrollo y el equipo de pruebas. Luego de una autoevaluación reciente, sabemos que el grado de madurez que han alcanzado en la implementación de la metodología SCRUM es del 56%, siendo Equipo y Pruebas las áreas de la agilidad con menos madurez como se muestra en el siguiente gráfico.

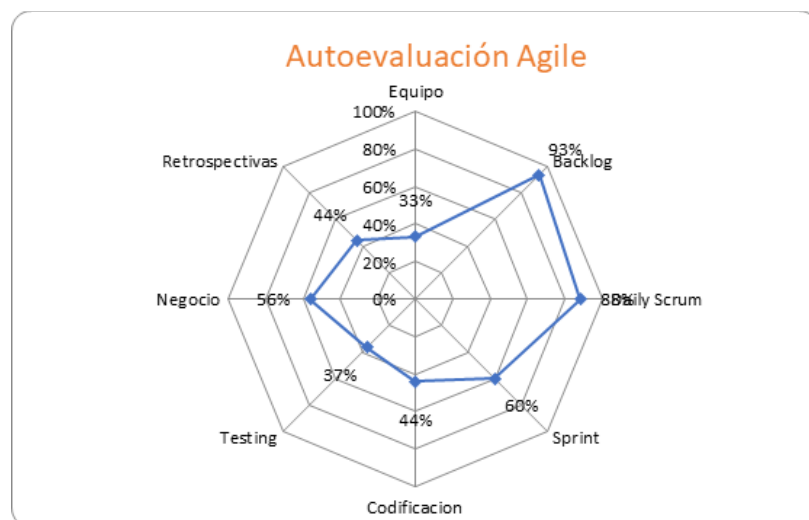


Figura 1.1 Autoevaluación Agile

De la evaluación anterior también hemos podido identificar que los siguientes problemas se presentan con frecuencia:

1. El equipo de pruebas completamente separado del proceso de desarrollo.
2. Se ha incrementado el tiempo de ejecución de pruebas por desarrollos incrementales.
3. Desconocimiento de las actividades a ejecutar por parte del equipo de pruebas con respecto a la metodología Scrum.

En equipos SCRUM, el trabajo de pruebas no es ejecutado únicamente por un equipo de pruebas sino también por todos los miembros del equipo SCRUM en una labor conjunta y no hay detalle del proceso a realizar cuando existe un equipo de pruebas.

Sabiendo que para la empresa XYZ la calidad del producto y rapidez de desarrollo son los puntos más fuertes de su estrategia competitiva, buscamos una forma de implementar mejores prácticas de metodologías ágiles que le permitan a la empresa aumentar su grado de madurez respecto a las mismas de una forma que sea fácil de manejar para el equipo y de esta manera mejorar los aspectos que afectan la parte estratégica de la compañía.

1.3 SOLUCIÓN DEL PROBLEMA

Para poder implementar mejoras al Subproceso de pruebas del proceso de desarrollo de software vamos a utilizar la metodología de Gestión de Procesos

BPM. El BPM es una metodología cuyo objetivo es mejorar el desempeño y optimizar los procesos en una empresa alineando los procesos al negocio y siendo la empresa XYZ una empresa de soluciones informáticas uno de los procesos principales es justamente el desarrollo de software.

La solución propuesta es orientada a las fases del BPM donde realizaremos el análisis de la situación actual del proceso Scrum llevado por el área de desarrollo y nos enfocaremos en las actividades del subproceso de Pruebas, realizando el modelo As-IS para tener el estado actual del proceso.

Realizaremos la revisión de las mejores prácticas de Agile Testing con el fin de evaluar cómo podemos enfocar al equipo de pruebas en las actividades que deben realizar y su rol dentro del equipo Scrum.

Los principios del Agile Testing van de la mano de los principios de las metodologías ágiles, creando la integración continua entre el equipo de pruebas y el equipo de desarrollo, permitiendo alta calidad en el producto entregado.

Integraremos las mejores prácticas de Agile Testing dentro del proceso de la metodología Scrum, de esta manera, el equipo de pruebas se acoplará al equipo Scrum y tendrá claras las actividades a realizar dentro del mismo

Una vez definidas las mejoras al Subproceso de pruebas, se utilizará un modelo descriptivo de BPM To-Be, el mismo que detallará las actividades y cambios a realizar durante el subproceso de pruebas del proceso de desarrollo de software.

Para comprobar el resultado de las mejoras propuestas, se realizará una evaluación del subproceso de pruebas mejorado, implementando el proceso mediante un plan piloto realizado en dos Sprints.

1.4 OBJETIVOS ESPECÍFICOS

Los siguientes son los objetivos específicos de este proyecto:

1. Revisar el subproceso de pruebas actual dentro del proceso de desarrollo de software mediante el modelo BPM As-IS.
2. Determinar las mejoras que pueden aplicarse al subproceso siguiendo la guía de mejores prácticas de la metodología Agile Testing.
3. Modelar el subproceso incluyendo las mejoras propuestas mediante el modelo BPM To-Be.
4. Analizar los resultados de las mejoras propuestas en el subproceso de Pruebas.

1.5 METODOLOGÍA

El enfoque, de este proyecto es el de un estudio de tipo cuantitativo que pretende medir y comparar el subproceso de pruebas actual dentro del proceso de desarrollo de software de la empresa contra una propuesta de mejora.

Para proponer una mejora al subproceso de pruebas se realiza una revisión de literatura con respecto a Agile Testing y las mejores prácticas que se puedan incluir en el subproceso sin causar mayores complicaciones a los equipos involucrados.

Para definir el alcance del proyecto consideramos como nuestro objeto de trabajo principal el subproceso de pruebas dentro del proceso de desarrollo de software de la organización. Los datos, históricos que se utilizan corresponden a un único proyecto dividido en 20 sprints que corresponden al trabajo realizado en los años 2019 y 2020. Los datos resultantes de la implementación corresponden a 2 sprints del proyecto antes mencionado ejecutados en el año 2021.

Las siguientes variables se han definido como las variables de interés del proyecto:

- Defectos reportados por los clientes:
 - Definición conceptual: Cantidad de defectos reportados por clientes.
 - Definición operacional: Conteo de la cantidad de defectos que han sido reportados por los clientes al equipo de servicios y que han sido validados por el equipo de pruebas como defectos.
- Cantidad de puntos trabajados en el sprint:
 - Definición conceptual: Según lo definido por la organización, el trabajo en diferentes módulos del sistema tiene diferentes pesos que son asignados en forma de puntos. La cantidad de puntos hace referencia a la cantidad de trabajo que se realizó en un módulo específico considerando los pesos según su complejidad.
 - Definición operacional: Suma de puntos que se trabajaron en el sprint, multiplicados por el peso asignado por la organización a

cada punto utilizando una tabla previamente definida. Los pesos se mantienen constantes tanto en los datos históricos de los años 2019 y 2020 como en la actualidad.

La investigación se realiza a través de un modelamiento y análisis del subproceso de pruebas actual, en el cual se buscan los puntos críticos en donde se enfocan las mejoras. Se realiza un análisis de mejoras prácticas de Agile Testing y se define un criterio de selección que se utiliza para escoger las que se han considerado más importantes y cuya implementación se considera factible en el corto plazo en relación al problema planteado y al contexto de la empresa. Teniendo definidas las propuestas de mejora, se definen también las métricas, basadas en los datos anteriormente recopilados, que usamos para tener resultados objetivos. Considerando las mejoras propuestas, se realiza un rediseño basado en BPM del subproceso de pruebas.

La estrategia de recopilación de datos involucra el repositorio central que ya existe en la organización del cual obtenemos el registro de datos de defectos reportados por clientes y de puntos trabajados en cada sprint. Estos datos son sumados y se preparan para su uso mediante métricas definidas.

El proceso rediseñado se implementa a modo de prueba en el trabajo regular de la compañía haciendo uso de dos sprints con lo cual se obtienen los datos resultantes de las métricas definidas. Finalmente se hace una comparación cuantitativa con la situación regular de la compañía de donde se presentan los resultados y conclusiones.

CAPÍTULO 2

2 MARCO TEÓRICO

2.1 METODOLOGÍA DE GESTIÓN DE PROCESOS (BPM)

Administración de procesos de negocio (BPM por sus siglas en inglés) es una disciplina de administración en la que los procesos del negocio son tratados como bienes y se basa en que la organización puede alcanzar sus objetivos a través de la definición, diseño, control y mejora continua de los procesos del negocio. [1]

La idea principal de BPM es enfocarse en los procesos al momento de administrar el trabajo de una organización. Su ciclo de vida se ve representado por la figura 2.1.

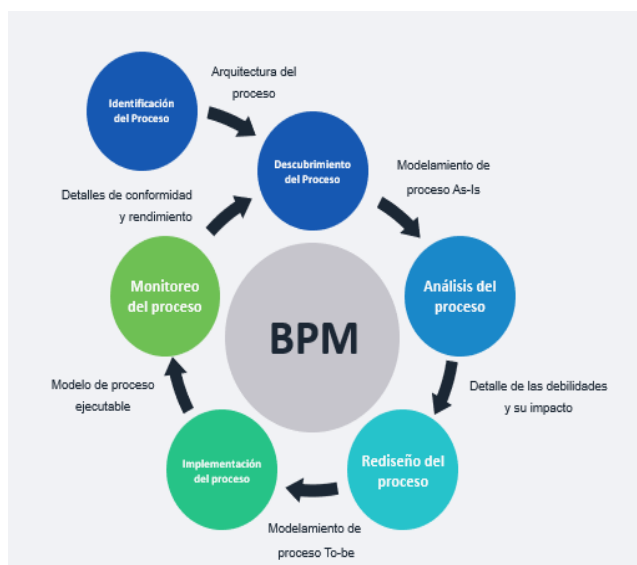


Figura 2.1 Ciclo de Vida BPM

El ciclo de vida de BPM se puede dividir en las siguientes etapas:

- **Identificación del proceso.** Se presenta un problema del negocio y se identifican los procesos que están relacionados al mismo. En esta etapa se deben establecer los límites de cada proceso y sus relaciones con otros procesos. Habiendo hecho esto, queda definida una arquitectura de donde se selecciona qué procesos van a ser administrados a lo largo del ciclo de vida.
- **Descubrimiento del proceso.** A través de un proceso de modelamiento conocido como As-Is se define cual es el estado actual de los procesos de la organización.
- **Análisis del proceso.** Se debe identificar cuáles son los problemas del proceso actual y de ser posible hacer mediciones cuantitativas que muestren cuál es el rendimiento de estos. Se debe establecer una

prioridad para cada uno de los problemas identificados tomando como base cual es el impacto de cada uno y el esfuerzo que requeriría resolverlos.

- Rediseño del proceso. Se deben plantear cambios en el proceso actual que sirvan para contrarrestar los problemas identificados en la etapa de análisis del proceso. Típicamente se van a considerar varias propuestas de mejora y se seleccionarán las que se considere puedan tener un mayor impacto positivo y ayuden a la organización a conseguir sus objetivos de mejor forma. Se genera un modelo con el proceso mejorado que se conoce como modelo To-Be.
- Implementación del proceso. Teniendo ya el modelo To-Be listo, la implementación del proceso implica adoptar el mismo como proceso activo, involucrando a todos los que tienen relación con el proceso y cambiando su forma de trabajar respecto al mismo.
- Monitoreo del proceso. Con el rediseño implementado, se deben utilizar métricas para poder evaluar el rendimiento del proceso considerando los objetivos planteados. A medida que el proceso rediseñado es utilizado pueden aparecer nuevos problemas y potenciales mejoras por lo que tenemos un ciclo de mejora continua. [2]

2.2 MARCO DE TRABAJO SCRUM

Scrum es un marco de trabajo de buenas prácticas para desarrollar, entregar y mantener productos complejos; estas prácticas no son nuevas, se vienen dando desde los años 80 pero en los 90 fue Ken Schwaber y Jeff Sutherland los que

presentaron por primera vez sus experiencias y las mejores prácticas en la industria. El Scrum se centra en entregas parciales y regulares del producto final comenzando por aquellas funcionalidades más importante para los clientes. [3]

El marco de trabajo Scrum consiste en el equipo Scrum y sus roles, eventos, artefactos y reglas relacionadas entre sí que permiten la organización empírica y dinámica del producto que se está desarrollando. Para que se use Scrum debe ser un proyecto que permita iterarse y entregar de manera incremental partes funcionales.

El equipo Scrum

El equipo Scrum se encuentra formado por los siguientes roles: El Dueño del producto (Product Owner), el equipo de desarrollo (Development Team) y un Scrum Master. Las características principales de este equipo es que son Autoorganizados y multifuncionales. Autoorganizados porque sin la supervisión de nadie realizan su trabajo de forma organizada y multifuncionales porque todos tienen las competencias necesarias para realizar un trabajo.

El Product Owner

Es el responsable que el equipo de desarrollo entregue el máximo valor del producto de acuerdo con las necesidades enlistadas en el product backlog, asegurando que el equipo de desarrollo comprenda claramente cada elemento.

El equipo de Desarrollo

Son los profesionales que realizan el trabajo de entregar un incremento del producto "Terminado". Este equipo debe ser de menos de 9 personas para que sea fácil de coordinar entre ellos.

Se considera como equipo de desarrollo a los profesionales que trabajan juntos en el desarrollo incremental del producto sin que se realice distinción de título, ni se reconoce a sub-equipos como pruebas, arquitectura; cada miembro del equipo tiene una habilidad especializada pero la responsabilidad recae en el equipo de desarrollo como un todo.

El Scrum Master

Es el responsable de promover y facilitar el Scrum a el equipo de desarrollo, el product owner y a la organización, ayudando a que comprendan las reglas, teorías, prácticas y valores del Scrum.

2.3 PROCESO SCRUM

En el siguiente gráfico se resume el proceso Scrum identificando los principales hitos.

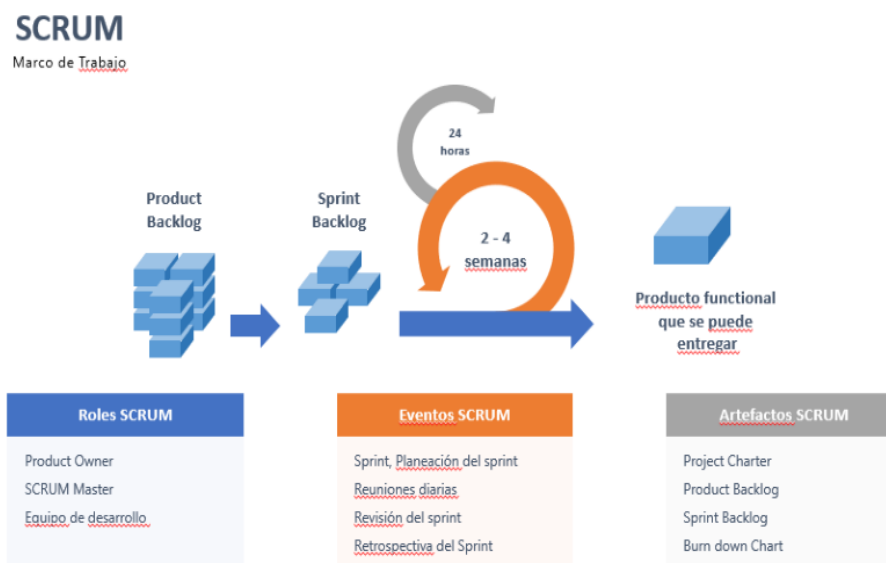


Figura 2.2 Proceso Scrum

El Product Owner organiza la lista de tareas que se encuentra en el Product backlog, en esta lista se especifica en alto nivel las funcionalidades recolectadas de las necesidades del usuario, priorizadas de acuerdo con el valor que tenga hacia al cliente.

Una vez organizada y priorizada, se realiza la reunión de planificación del sprint, donde se crea el plan mediante el trabajo colaborativo del equipo enfocados a qué puede entregarse dentro del mismo y cómo se conseguirá realizar ese trabajo. El resultado de esta planificación es el sprint backlog junto con los objetivos del sprint.

Luego inicia el sprint, que es la parte central del Scrum, tiene como máximo un tiempo de duración de 4 semanas y es donde el equipo scrum desarrolla la implementación de las funcionalidades. Durante el Sprint se realizan las

reuniones diarias de máximo 15 minutos donde todo el equipo de desarrollo planea el trabajo de las próximas 24 horas, evalúan el progreso y exponen los impedimentos para cumplir los objetivos del sprint. El trabajo de desarrollo se centra en cumplir los objetivos de mayor valor y se puede negociar un nuevo alcance evitando realizar cambios a los objetivos.

En cualquier momento es posible realizar la suma del trabajo restante para alcanzar el objetivo del sprint y así se puede obtener la proyección de la tendencia del trabajo pendiente (burndown chart), completo y flujo acumulado que es presentado al equipo en las reuniones diarias para que todos sepan cómo se va avanzando o las posibles acciones a tomar para cumplir el objetivo.

El término "Done" (Terminado) es cuando el equipo define cuando se termina el script. Una vez terminado la implementación y pruebas se realiza la reunión de revisión del Sprint donde el equipo scrum y los interesados colaboran en lo que se hizo bien, en lo que hay que mejorar y lo que se debe mantener para aplicar en las siguientes iteraciones.

Finalmente se realiza la retrospectiva del sprint en donde se busca la oportunidad para que el equipo Scrum se inspeccione a sí mismo y se cree un plan de mejora para el siguiente sprint en cuanto a personas, el proceso y las herramientas. El scrum master alienta al equipo a mejorar dentro del marco del Scrum. [4]

2.4 AGILE TESTING

Fundamentos del Agile Testing

¿Qué es Agile Testing?

Es un conjunto de prácticas de pruebas que se realizan desde el inicio hasta la entrega y posterior a ella, enfocándose en dar un producto de valor al cliente. Las actividades que se realizan están orientadas a la obtención de calidad del producto y en fortalecer y apoyar el pensamiento que todo el equipo es responsable de la calidad. [5]

Los 10 principios del Agile Testing

Para que un equipo sea ágil, no es solo con que cada uno de sus miembros sean más rápido, es cuando su mentalidad de trabajo en equipo, equipo autoorganizado da mejores resultados. Todo equipo ágil debe seguir los lineamientos del manifiesto ágil y es de aquí que nace los principios del agile testing.

Los siguientes principios no necesariamente son para un equipo de pruebas, es para cualquier miembro del equipo que desea liberar productos de alta calidad.

Son:

- Proveer continua retroalimentación
- Entregar valor al cliente
- Tener comunicación frente a frente con el cliente
- Tener coraje

- Pensar Simple
- Practicar la mejora continua
- Responder al cambio
- Autoorganizarse
- Centrarse en las personas
- ¡Disfrutar!

Testing Manifiesto

Así como existe el manifiesto ágil, este manifiesto es creado con un enfoque para el cambio de mentalidad al Tester/Qa en un entorno ágil. Este fue creado por Karen Greaves y Samantha Laing [6]. Se centra en indicar que las pruebas se realizan en todo el proceso de desarrollo, enfocándose en prevenir los defectos y que todo el equipo es responsable de la calidad, indicados de la siguiente manera:

Tabla 1 Testing Manifiesto

Testing Manifiesto		
Probando durante todo el desarrollo	Por encima de	Probar al final
Prevenir errores	Por encima de	Encontrar errores
Probar el arreglo	Por encima de	Comprobación de la funcionalidad
Construir el mejor sistema	Por encima de	Dañar el sistema
El equipo es responsable de la calidad	Por encima de	El tester es el responsable de la calidad

Contexto de Roles y Actividades en un equipo ágil

Para poder comprender las pruebas en un entorno ágil, de manera general, vamos a compararlas con el método tradicional.

En la figura 2.3 se muestra dónde se encuentra la ejecución de pruebas utilizando el método cascada. El equipo de pruebas interviene en la ejecución casi al final del proyecto, donde este gráfico sería el más óptimo si no existieran errores, incluso algunos graves, que involucran cambiar los requerimientos y por ende la codificación.

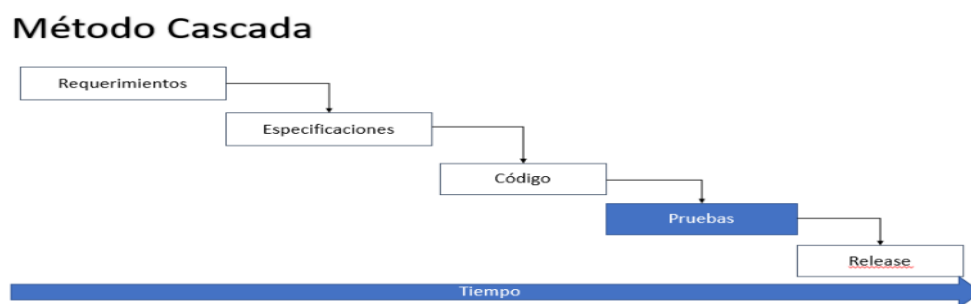


Figura 2.3 Pruebas en método Cascada

En cambio, en ágil el proceso es iterativo e incremental, como se muestra en la figura 2.4, lo que significa que pruebas interviene una vez que la codificación de esa iteración está realizada y cada parte de la codificación es probada antes de pasar a la siguiente iteración.

Agile

Iterativo e incremental

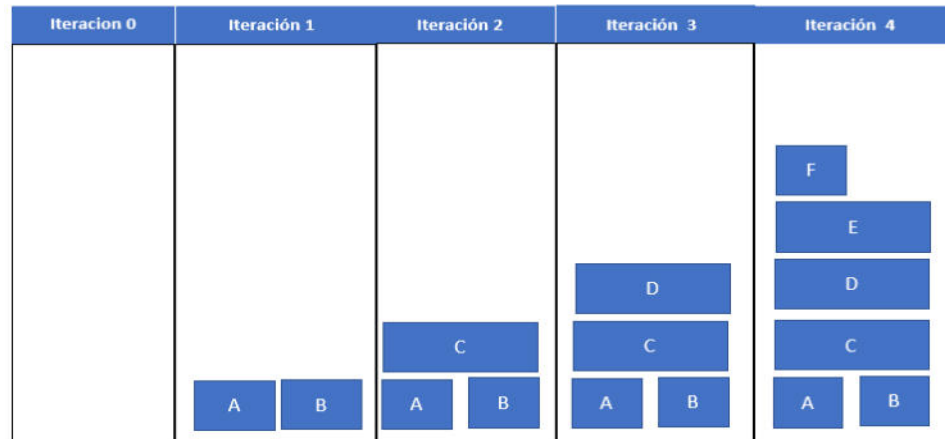


Figura 2.4 Pruebas en metodologías ágiles

Pruebas es una parte central dentro del equipo ágil, ellos forman una parte del equipo de cliente y equipo de desarrollo, como se muestra en el gráfico 5.

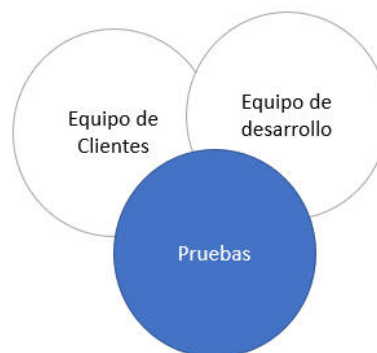


Figura 2.5 Interacción de roles

El equipo de Clientes son los expertos del negocio, los clientes, los product owner, los jefes de productos, analistas de negocio, todo que tenga relación con el negocio dentro del proyecto. Pruebas son parte integral del equipo cliente

ayudando al equipo del cliente en la definición de las historias de usuario, en definición de los ejemplos y los criterios de aceptación.

El equipo de desarrollo es todos los que liberan código, programadores, administradores, arquitectos, administradores de base de datos. Pruebas también es parte del equipo de desarrollo porque las pruebas es un componente central dentro del equipo ágil, ellos abogan por lo calidad a nombre del cliente y ayudan a desarrollo a dar el máximo valor al cliente.

PROCESO DEL AGILE TESTING

Como se muestra en la figura 2.6, la participación del Tester es en todas las etapas del proceso de desarrollo ágil. Las actividades van más allá de solo encontrar defectos y probar funcionalidad en las aplicaciones.

Como hemos visto el rol de Tester en Scrum no existe, normalmente es realizado por el mismo desarrollador o por el Product Owner; pero algunos proyectos Scrum tienen equipos de pruebas dedicados.

Cuando se inicia un proyecto que va a tener algunas iteraciones, el enfoque de pruebas también es incluido, con el fin de tener una visión global de todo el proyecto.

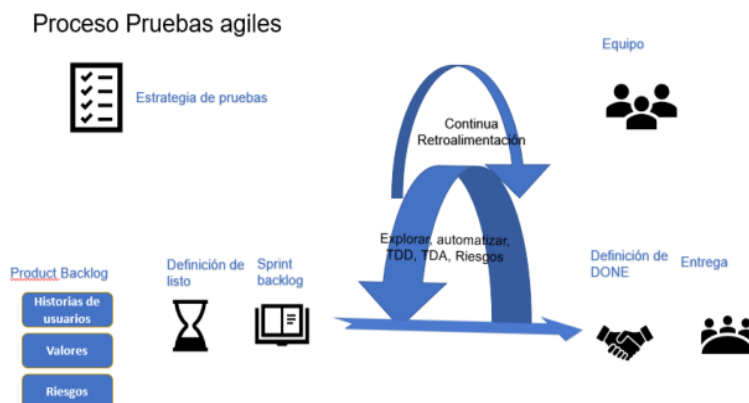


Figura 2.6 Proceso de pruebas ágiles

Lo principal es conocer en etapas tempranas las estrategias de pruebas que se deben aplicar desde el punto de vista de todo el proyecto de manera muy general. Lo importante es tener una visión global del plan y que todo el equipo tenga conocimiento de la estrategia de pruebas para todo el proyecto. [7]

Todo el equipo define cuando cada iteración está lista. La definición de DONE es muy importante para que todo el equipo esté de acuerdo con los criterios y acuerdos para cuando deba concluir las pruebas unitarias, pruebas de integración, pruebas de sistema, las historias de usuarios, las características del sistema, la iteración, el reléase.

Cuando ya se ingrese a la planificación del Sprint las actividades de pruebas se deben de enfocar definir con el equipo las historias de usuarios a trabajar en el sprint, que las historias estén completas y tengan ejemplos necesarios para la certificación de estas. Definir las estrategias de pruebas a realizar en el Sprint, que tipos de pruebas realizar con que datos, con qué recursos, necesidades de capacitación, etc. Priorizar defectos que se encuentren por iteraciones

anteriores y validarlas con el cliente para incluirlas. Como vemos es, todo lo que necesito tener antes de empezar el sprint.

Durante el sprint se realiza la codificación y las pruebas, es aquí que se inicia a escribir al detalle las pruebas a realizar, se identifica las historias de usuarios que se van a realizar pruebas exploratorias y las verificaciones de la codificación. Es aquí donde se debe trabajar estrechamente con desarrollo, en pruebas de pares, “show me”, realización de TDD, ATTD, etc. Lo más importante en este punto es que el Tester no se quede esperando que le llegue algo, debe ser proactivo mostrando al cliente las características avanzadas, considerando los defectos que se deben solucionar en el sprint lo que pueden quedar como deuda técnica.

Durante las reuniones diarias, se comunican los problemas desde el punto de vista de pruebas, las tareas que se trabajarán en el día, visualizar las tareas, los impedimentos que se puedan dar.

Cuando el equipo ya tiene pruebas de regresión automatizadas, la idea es que en esta etapa las reutilice y sean aplicadas para cubrir historias de usuarios de iteraciones anteriores.

Por otra parte, durante el sprint los desarrolladores deben ejecutar las pruebas unitarias automatizadas antes de verificar el código nuevo. Si existe falla en estas pruebas, la prioridad más alta del equipo (aparte de un problema en producción) debe ser solucionarlo y hacer que la compilación funcione nuevamente.

Durante la iteración, se están automatizando las nuevas pruebas. Tan pronto como pasen se agregan a las suites de regresión. A medida que se completan las historias de usuarios, y se confirman su funcionalidad deben incluirse en el paquete de regresión. Las pruebas de regresión deben estar controladas y versionadas. Pero mientras no se tenga todo o la mayor parte automatizada, se puede realizar las pruebas exploratorias manuales, hasta que las historias de usuarios se completen.

También es muy importante ir midiendo como va avanzando el equipo con los gráficos de evaluación de la iteración. Es importante ir midiendo la cantidad de pruebas ejecutadas por historia o por área funcional, el estado de la automatización, el número de pruebas pasadas/fallidas en el tiempo, el resumen de historias, métricas de defectos.

Luego de completar una iteración, la retrospectiva ayuda al equipo a mejorar las acciones que consideren las oportunidades de mejora y se aprovecha para identificar obstáculos relacionados con las pruebas.

Cuando el reléase ha finalizado con éxito y pasando las pruebas dentro de la iteración, es hora de realizar las pruebas de aceptación de usuarios, lo que significa es aceptación de una liberación. Las UAT son realizadas por los usuarios verificando la usabilidad del sistema y verificando la funcionalidad con sus propios datos. Aquí los Testers brindan el soporte a los clientes de las pruebas ejecutadas, el registro de los defectos hasta su finalización.

CAPÍTULO 3

3 ANÁLISIS Y LEVANTAMIENTO DEL PROCESO ACTUAL

3.1 MODELO MACRO DEL PROCESO DE DESARROLLO

En este capítulo analizaremos la situación actual de la empresa XYZ, los equipos que la conforman y cómo realizan el proceso de desarrollo de software de manera macro y al detalle el proceso del equipo de pruebas.

La empresa tiene tres áreas definidas para el desarrollo y soporte de nuevos productos de software que se ofrecen a los clientes actuales o se vende a nuevos clientes por medio de licencias. Estos equipos son: El equipo de Desarrollo, equipo de pruebas y el equipo de Cliente, como se indica en la siguiente figura 3.1.

Equipo de Desarrollo	Equipo de Pruebas	Equipo Cliente
<ul style="list-style-type: none"> • Formado por los desarrolladores • Existen desarrolladores líderes por producto, se especializan en un producto. • Los desarrolladores juniors trabajan ayudando a los líderes de productos. • Implementan las nuevas características de los productos. 	<ul style="list-style-type: none"> • Está formado por un líder QA • Cada analista de calidad se especializa en un producto. • Se asigna a los proyecto de acuerdo a sus capacidades y destrezas. • Validan la calidad del software de acuerdo con los requerimientos solicitados. 	<ul style="list-style-type: none"> • Se dividen en Product Owner, Soporte a clientes y consultores • El PO trabaja con el equipo de desarrollo y pruebas. • Las personas de soporte resuelven los incidentes reportados por los clientes • Los Consultores, trabajando directamente con el cliente en aplicar el sistema en sus empresas

Figura 3.1 Equipos proceso de desarrollo

Como hemos indicado, el proceso de desarrollo se lleva bajo el marco de trabajo SCRUM, por lo que los equipos interactúan como se muestra en la siguiente figura 3.2.

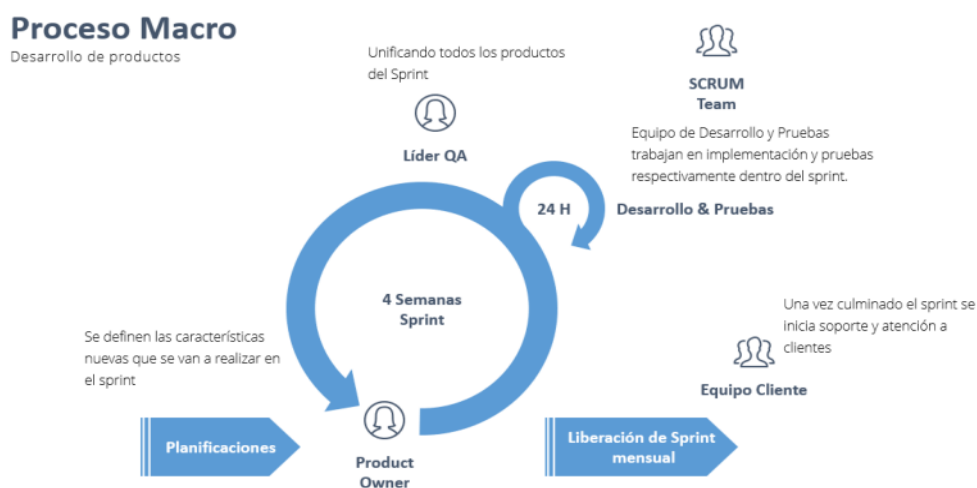


Figura 3.2 Proceso desarrollo de software

Al inicio de cada mes el Product Owner junto con el equipo de cliente identifican las características y errores de producción más prioritarias que se deben trabajar en el mes. Se hace la reunión de inicio de mes explicando todas las necesidades que deben realizarse en el sprint y sus respectivas prioridades.

Con esta información el equipo de desarrollo define los tiempos de implementación y el equipo de pruebas define los tiempos de pruebas. Las características que no ingresan en el mes se quedan en el Product Backlog.

Se realiza el desarrollo y una vez concluido se realizan las pruebas de lo implementado. Luego existe una etapa de solución de errores de las pruebas realizadas hasta que se encuentre lista la aplicación para ser liberado el sprint.

Una vez concluido el sprint se indica al equipo cliente la liberación de las nuevas características y errores solucionados para que ellos informen a los clientes e instalen la nueva versión a los diferentes clientes que la deseen.

3.2 MODELO AS-IS SUBPROCESO DE PRUEBAS

El alcance que hemos definido para nuestro proyecto enfoca el trabajo únicamente en el subproceso de pruebas y este se ha dividido en tres etapas o hitos que son:

Tabla 2 Color referencia de Etapas

Etapas	Color Referencial*
Planificación y estimación	
Implementación	
Revisión del Sprint	

* El color referencial se utiliza para ubicar a las tareas relacionadas a cada etapa en los modelos gráficos.

Basándonos en el ciclo de BPM, en este punto hacemos referencia a la identificación y descubrimiento del proceso en dónde por cada etapa hacemos una revisión de las actividades que actualmente se desarrollan.

La etapa de planificación y estimación empieza con la recepción de los requerimientos planteados por el Product Owner que pasan al equipo de pruebas para realizar la medición de historias de usuario como actividad inicial. Una vez realizada la medición de historias de usuario, se definen las prioridades de estas y se procede a documentar los casos de prueba que se van a utilizar y se da paso al inicio de la etapa de implementación.

La etapa de implementación empieza con la preparación de los ambientes de prueba que involucra a toda la infraestructura necesaria, incluyendo la preparación de los datos, para poder trabajar en las pruebas. Con los ambientes listos, el equipo de pruebas recibe del equipo de desarrollo un "Build" (producto) sobre el cual se ejecutarán las pruebas.

El equipo de pruebas avanza con la ejecución de estas, en donde, al momento de encontrar defectos o errores, estos son reportados para conocimiento del equipo de desarrollo. En el caso de que estos defectos sean críticos e impidan poder seguir con el proceso de pruebas regular, el proceso de pruebas se detiene y se comunica al equipo de desarrollo que hay una necesidad de corrección inmediata. El equipo de desarrollo debe trabajar lo más pronto

posible en corregir el error y enviar la corrección para que el equipo de pruebas pueda confirmar que este problema crítico ha sido solucionado. Se considera en la actualidad un promedio de dos horas para resolver un error crítico. En el caso de que los defectos no sean críticos y no impidan la realización del proceso de pruebas, el equipo de pruebas los va a listar y detallar para poder enviar al equipo de desarrollo los defectos del día que podrán ser trabajados en los siguientes dos días.

Estas actividades se repiten a manera de ciclo hasta que se terminan las pruebas tal como se han planificado y se da paso a la etapa final de revisión del Sprint.

En la etapa de revisión del Sprint, el Sprint es liberado por el equipo de pruebas y está disponible para su uso en ambientes de producción. Con esto, se establecen reuniones que incluyen al Product Owner y los equipos de desarrollo y pruebas, para realizar la retrospectiva y revisión del Sprint con el fin de obtener información accionable que puede servir para mejorar las futuras iteraciones.

Un modelo BPM que describe lo anteriormente se muestra en el Anexo 1.

3.3 ANÁLISIS DEL MODELO

Siguiendo con el ciclo BPM, en este punto hacemos referencia a la etapa de análisis del proceso. Para analizar el modelo actual vamos a utilizar una técnica de tipo encuestas para cuantificar el criterio de los actores involucrados en el desarrollo del mismo respecto a temas críticos en diferentes etapas del subproceso.

Los participantes de la encuesta serán 10 personas que han sido escogidos de la siguiente forma:

- 2 miembros el equipo cliente (que tiene contacto directo con los clientes)
- 3 miembros del equipo de pruebas.
- 3 miembros del equipo de desarrollo
- 2 miembros de la directiva de la empresa

El cuestionario incluye 3 etapas que hacen referencia a los hitos que se alcanzan en el subproceso de pruebas.

- Planificación y estimación
- Implementación
- Revisión del Sprint

Las preguntas que se han efectuado a los participantes se han definido de la siguiente forma:

- ¿Existen demoras en el proceso?
- ¿El proceso tiene disponibles muy pocos recursos?
- ¿Se genera frustración entre los participantes?
- ¿Existe mucho trabajo manual?
- ¿Se generan resultados de mala calidad?

Los resultados numéricos de la encuesta se pueden ver en la tabla 3 a continuación.

Tabla 3 Resultado de encuesta

Preguntas	Planificación y estimación		Implementación		Revisión del sprint	
	Si	No	Si	No	Si	No
¿Existen demoras en el proceso?	4	6	6	4	2	8
¿El proceso tiene disponibles muy pocos recursos?	6	4	7	3	1	9
¿Se genera frustración entre los participantes?	8	2	9	1	3	7
¿Existe mucho trabajo manual?	2	8	9	1	5	5
¿Se generan resultados de mala calidad?	9	1	8	2	5	5

Para visualizar mejor los resultados los presentamos en la figura 3.3 en donde podemos notar que la distribución de las respuestas tiene un sesgo hacia la clase en donde el "Sí" representa que se percibe o evidencia un problema especialmente en las etapas de preparación y pruebas.

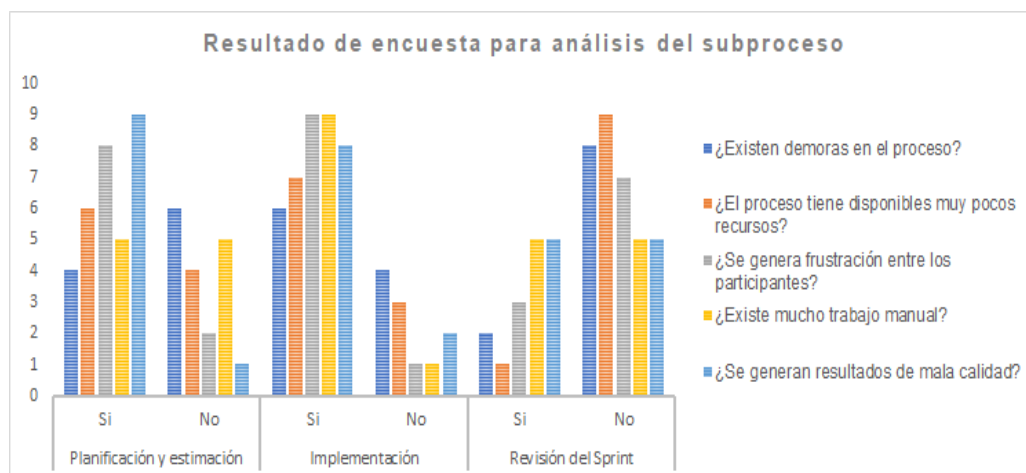


Figura 3.3 Resultado de encuesta

Separando por etapa, podemos notar que en la etapa de preparación los puntos más críticos que arroja la encuesta están alrededor de la generación de resultados de mala calidad, la frustración de los participantes y los recursos disponibles para esta parte del subproceso.

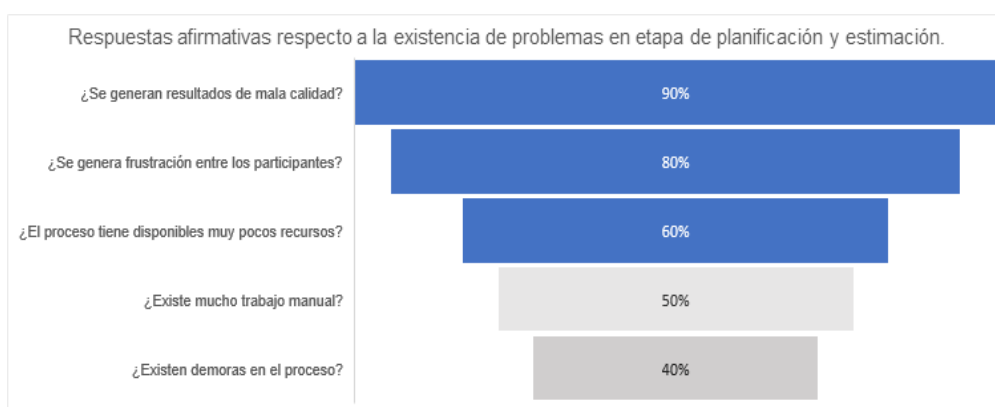


Figura 3.4 Respuestas afirmativa etapa planificación.

En la etapa de pruebas, todas las preguntas fueron contestadas en una mayoría de forma afirmativa lo que indica la presencia de problemas siendo los puntos

más críticos la frustración entre participantes, el exceso de trabajo manual y los resultados de mala calidad.

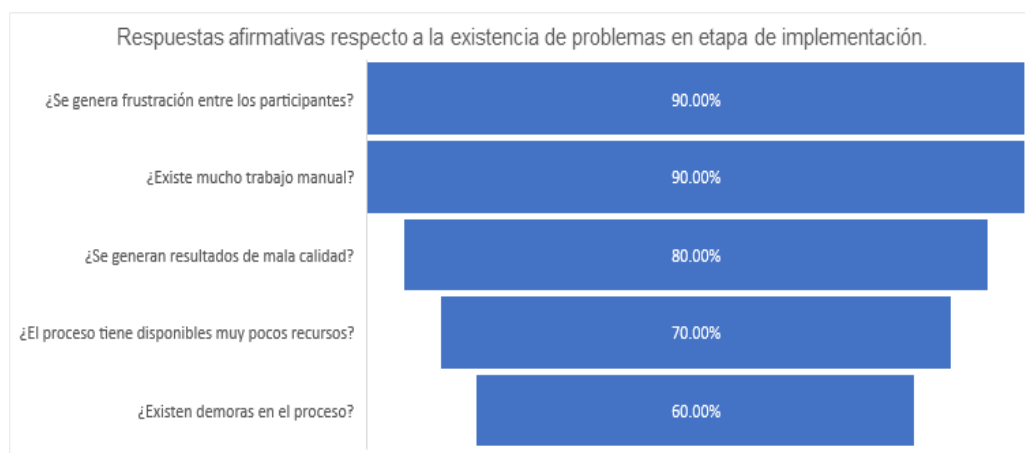


Figura 3.5 Respuestas afirmativas etapa implementación.

En la etapa de entrega, vemos que las respuestas están un poco más balanceadas que en las etapas anteriores y no se evidencia un punto en el que los problemas se consideren en un estado crítico en base a las respuestas de la encuesta.

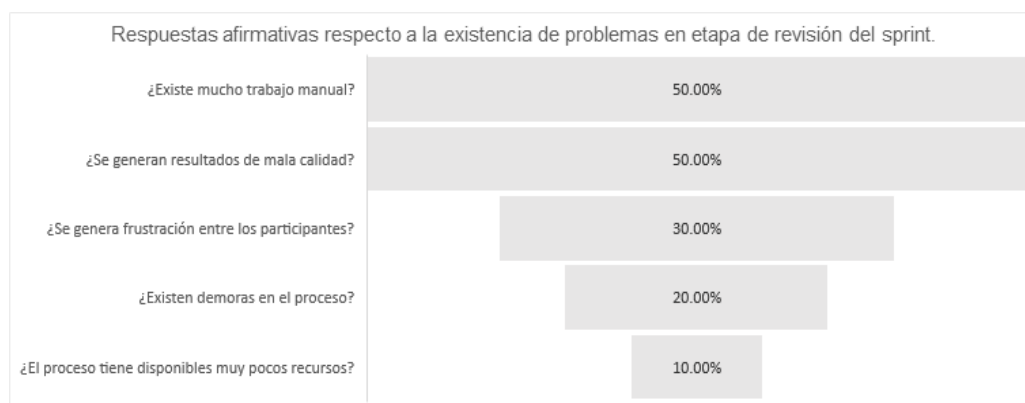


Figura 3.6 Respuestas afirmativas etapa de cierre.

Finalmente, orientando el enfoque hacia las preguntas que en su mayoría muestran la existencia de problemas, les pedimos a los encuestados que listen algunos de los problemas más críticos asociados a cada etapa de donde recogimos los siguientes puntos:

Planificación y estimación

- No se incluye a todos los miembros del equipo de pruebas para el diseño del producto.
- No se diseñan las estrategias de pruebas en conjunto entre los equipos de desarrollo y pruebas.
- Falta de comunicación con el equipo cliente

Implementación

- No se realizan pruebas unitarias en el desarrollo.
- Se da paso al proceso a pruebas solo cuando el producto está terminado.
- Poco tiempo disponible para realizar las pruebas antes de que termine el sprint.
- No existe automatización de casos de pruebas.
- Existe gran cantidad de errores durante las pruebas.

Revisión del Sprint

- No se realizan pruebas de aceptación.
- No se realiza un proceso de puesta en producción.

De forma cuantitativa y basándonos en los los datos históricos recopilados en los años 2019 y 2020, el equipo de desarrollo compuesto por 9 colaboradores trabaja una media de 49 puntos en cada sprint. Estos puntos han sido previamente definidos por la organización de acuerdo con pesos asignados según la complejidad que implica desarrollar un tipo de funcionalidad.

De estos datos también podemos observar que la media de defectos reportados por los clientes en cada Sprint es de aproximadamente 7 defectos. Es importante recalcar que la empresa desarrolla una solución que tiene como una de sus mayores ventajas competitivas ser de alta flexibilidad, lo que permite que esta sea utilizada por una cantidad de clientes con diferentes modelos de negocio. Esto presenta también un reto para la empresa ya que los defectos que se presenten pueden tener gran impacto en el negocio de los clientes.

CAPÍTULO 4

4 ANÁLISIS DE MEJORES PRÁCTICAS DE AGILE TESTING

4.1 ANÁLISIS DE ESTRATEGIAS DE AGILE TESTING

Siendo que Agile Testing es la práctica de pruebas de software en un entorno ágil que tiene un grupo de guías y prácticas que van a ayudar al equipo de pruebas a mejorar la calidad del producto y a satisfacer las necesidades de los clientes, en este capítulo revisaremos esas actividades que recomienda agile Testing y la comparemos con lo actualmente realiza el equipo de pruebas. Esto nos permitirá evaluar cual sería el impacto de llevar esta práctica como mejora del proceso del equipo de pruebas en la empresa XYZ.

La calidad es responsabilidad de todo el equipo.

Al decir que la calidad es responsabilidad de todo el equipo, esto involucra el Product Owner, el desarrollador y el Agile Tester en el marco de trabajo scrum.

En el equipo scrum, el Agile Tester es la persona del equipo de pruebas que se encuentra en el centro con cara a la tecnología y con cara al negocio velando porque se cumpla con la calidad del producto realizando entrega de valor a un ritmo sostenible y continuo. [5].

En tabla 3 comparamos lo que agile Testing nos indica respecto a cómo debe ser el perfil del Tester y el impacto de la mejora de este perfil.

Tabla 4 Comparación Tester y Agile Tester

Tester	Agile Tester
No se involucra en el inicio del proyecto	Conoce de todo el proyecto desde el inicio
Los casos de pruebas no son revisados por los desarrolladores	Los casos de pruebas deben ser compartidos con los desarrolladores
No es parte de la reunión del levantamiento del requerimiento	Es parte de todas las reuniones desde el inicio de un requerimiento, cambio o nuevo proyecto.
No participa de las reuniones de definición de la arquitectura de la solución	Debe ser parte de la reunión de la definición de la arquitectura.
Se siente feliz de encuentra defectos	Se debe sentir feliz de prevenir los defectos
Las pruebas se realizan al final de la implementación	Realizar pruebas parciales del requerimiento, cambio o nuevo proyecto
Un error en producción la culpa es del Tester	Error en producción la falla es de todo el equipo.

El desarrollador, al ser la persona que realiza la codificación, su responsabilidad es diseñar y desarrollar un software con calidad. Al igual que el agile Tester, debe cambiar su forma de pensar con respecto a sus entregas y trabajar de la mano con el equipo. En la tabla se detalla los principales cambios y el impacto en el equipo de desarrollo.

Tabla 5 Comparación Desarrollador y Desarrollador en Agile Testing

Desarrollador Actual	Desarrollador en Agile Testing
Levantar los requerimientos solos	Hacer participe en todas las reuniones al Tester del proyecto
Conversar solo con los especialistas del producto	Hacer participe en todas las reuniones con los especialistas al Tester del proyecto
Mi responsabilidad no es hacer pruebas	Debo realizar pruebas unitarias completas para entregar un producto que se pueda probar
No hago automatización de pruebas	Realizar pruebas unitarias automáticas
No soy responsable de los errores en producción	Todo el equipo es responsable de la calidad del producto en producción
La arquitectura solo la conozco yo	Compartir la arquitectura con el equipo del proyecto

El product owner al ser el propietario del producto, es el que tiene claro cuál es el valor del producto frente a los clientes, por lo tanto, su contribución en la calidad y satisfacción al cliente es muy importante. En la tabla 5 se compara el product owner de la empresa XYZ con lo que indica la práctica y su impacto.

**Tabla 6 Comparación entre PO
y PO en Agile Testing**

El Product Owner	El Product Owner en Agile Testing
Informo cuál es la necesidad del producto	Estoy pendiente de cualquier inquietud sobre expectativas del cliente
Narro los requisitos del cliente	Indico todas las pruebas negocio que son necesarias realizar

Todos los roles deben pensar en:

- Ser proactivos en solución de problemas
- Velar por la entrega de valor al cliente
- Enfocarse en las pruebas de las características con más valor.
- Comprender los riesgos que se presenten desde los diferentes puntos de vista
- Todos pueden contribuir en los diferentes roles.

Por medio de la práctica llamada “Los tres amigos” [8] que ayuda a tener el enfoque de cada uno de los roles con el fin de ver desde las diferentes perspectivas un problema e intenta equilibrar la colaboración entre dichas personas.

Cuando se reúnen los tres amigos (negocio, desarrollo y pruebas) deben buscar:

- Identificar malentendidos y confusiones
- Entendimiento compartido de una entrega.
- Solución compartida de un problema.

Este grupo no solo se limita a tres personas, si son necesarias más, deberían incluirse en la reunión. Lo importante es decidir en equipo las soluciones ante cualquier problema.

Los Cuadrantes de Pruebas

Una de las principales prácticas que utiliza Agile Testing son los cuadrantes que pruebas. Como se muestra en la siguiente figura 4.1, cada cara del cuadrante refleja los diferentes tipos de pruebas que se pueden hacer en un entorno ágil. La división de una abscisa divide el cuadrante en las pruebas que soportan al equipo y las pruebas que critican el producto. La división de la otra abscisa divide en pruebas con cara al negocio y pruebas con cara a la tecnología. Cabe resaltar que los cuadrantes no es un orden que seguir para hacer las pruebas, sino una manera de contemplar lo que se necesita de pruebas en el proyecto bajo las diferentes visiones y tomar las que se necesiten en las estrategias de pruebas.

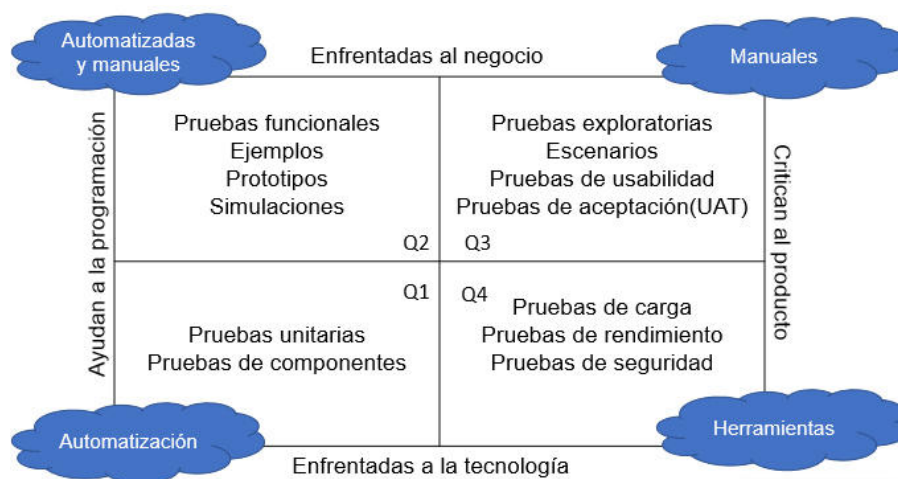


Figura 4.1 Cuadrantes de Agile Testing

En la empresa XYZ, el equipo de pruebas realiza los siguientes tipos de pruebas: Pruebas funcionales, pruebas de performance, pruebas de concurrencia, pruebas de cargas.

Lo que indica los cuadrantes de pruebas es tener un enfoque del tipo de pruebas a aplicar de acuerdo si apoyo a todo el equipo o criticando el producto.

Como se muestra en la figura 4.2 las pruebas que soportan al equipo es algo que no se realiza en el equipo de pruebas de la empresa XYZ ya que las pruebas se realizan una vez que está desarrollado. Lo que indica el cuadrante es ayudar al equipo de desarrollo mientras se está codificando por medio de estas técnicas y cubren los cuadrantes 1 y 2.

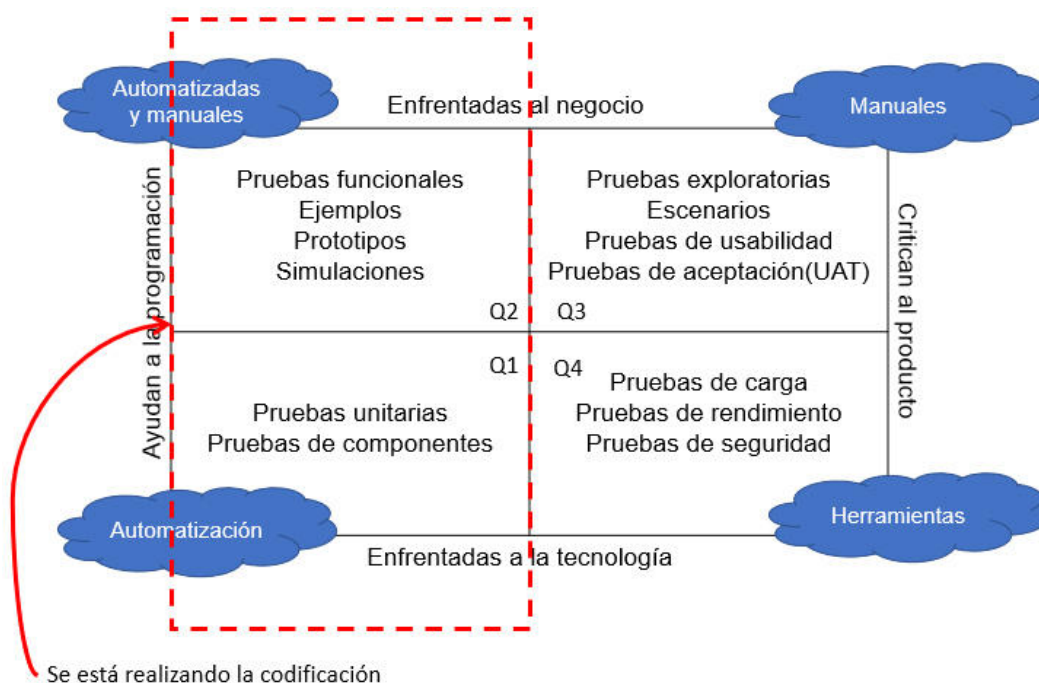


Figura 4.2 Cuadrantes con cara al apoyo al desarrollo

En el cuadrante 1 la idea principal es que el equipo de pruebas apoye con los diferentes casos de pruebas que se pueden presentar y los desarrolladores realicen las pruebas unitarias de cada objeto implementado y una vez codificados, realizar las pruebas de componentes con el fin de que la calidad se integre desde el inicio de la codificación. Lo ideal es realizar en este punto la automatización de las pruebas unitarias aplicando TDD (Test driven development – Desarrollo basado en pruebas).

Tabla 7 Tipos de pruebas cuadrante 1

Tipos de Prueba	Descripción	Aplica Equipo
Pruebas unitarias TDD - test driven development (Desarrollo dirigido por pruebas)	Es la realización de la codificación primero considerando la prueba (unitarias), luego escribir el código de programación y refactorizar lo codificado	NO
Pruebas de componentes	Prueba de un grupo de clases más grande que permite una prueba de un servicio.	SI
Pruebas unitarias Manuales	Prueba del componente que se está codificando	SI

En el cuadrante 2, a pesar de que sigue ayudando al equipo, esto es más orientado al negocio trabajando de la mano con el equipo de cliente (PO, analista de negocio, clientes) la idea es poder tener pruebas a un más alto nivel. Se realizan las pruebas de la funcionalidad buscando que se cumpla las necesidades del cliente, los especialistas del negocio se aseguran de que todo

esté claro con respecto a los diferentes escenarios de negocio que se presenten en la funcionalidad.

Tabla 8 Tipos de pruebas Cuadrante 2

Tipos de Prueba	Descripción	Aplica equipo
Herramientas para levantamiento de Ejemplos y requerimientos	El equipo debe ayudar al cliente a comunicar todos los requerimientos relacionados a las historias de usuario. Se pueden usar las siguientes herramientas: CheckList, mapas mentales, mock-up, diagramas de flujo, hojas de Excel. Estas permitirán completar las historias de usuario de manera correcta.	NO
Pruebas funcionales Manuales	Son las pruebas de la funcionalidad completa. Realizadas de manera manual, siguiendo los casos de pruebas analizados para la historia de usuario.	SI
Pruebas funcionales automatizadas	Comprende la automatización de las pruebas funcionales por medio de herramientas automatización orientadas a la captura del UI.	NO
Pruebas de regresión	Son las pruebas que verifican que lo anterior no se haya dañado por el nuevo incremento del cambio	SI
Pruebas de regresión automática	Las pruebas de regresión permiten desarrollar automáticamente las pruebas ahorrando tiempo en su ejecución.	NO
Pruebas de prototipo de interfaz con el usuario	Son pruebas que validan si el diseño de la interfaz está de acuerdo con las expectativas del cliente	NO

Las pruebas que critican al producto

La palabra “Crítica” no es en sentido negativo, lo que busca es dar sugerencias de mejora porque no es lo mismo verlo en un papel que la aplicación ya funcionando. Como se muestra en la figura 4.3 involucran los cuadrantes 3 y 4 una vez que se encuentra realizada la codificación.

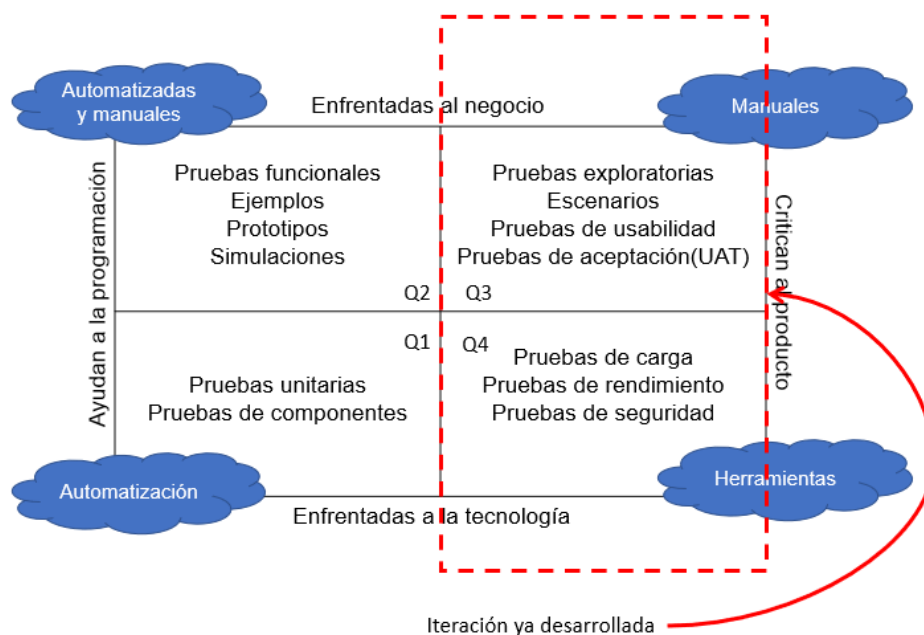


Figura 4.3 Cuadrantes con cara que critican el producto

El Cuadrante 3 la idea es comprobar que la funcionalidad cumple con las expectativas, la idea de estas pruebas son emular la forma en que el usuario final trabajaría con la aplicación.

Tabla 9 Tipos de pruebas del Cuadrante 3

Tipos de Prueba	Descripción	Aplica equipo
Pruebas Exploratorias	Son las pruebas donde se diseña, se prueba, se ejecuta y reporta defectos al mismo tiempo, para	NO

	una retroalimentación más rápida de los resultados esperados	
Pruebas de Usabilidad	Principalmente orientada a que el UI sea fácil de comprender y usar por parte del usuario	NO
Pruebas de Aceptación	Deben ser ejecutadas por el equipo del cliente, orientada a validar que se cumplan los criterios de aceptación de las historias de usuarios. Estas también pueden ser automáticas llamadas ATDD	NO

En el cuadrante 4 se encuentran las pruebas críticas no solo en un entorno ágil, sino en cualquier prueba de software. Lo principal de este cuadrante es tener en mente los requerimientos de rendimiento, seguridad, velocidad y otros atributos no funcionales antes de comenzar a codificar y pueden ser, incluso, mucho más importante que la funcionalidad real. No por estar en el último cuadrante es porque no sean importantes, deben considerarse en cada ciclo de desarrollo y no dejarse al final.

En muchos casos estas pruebas deben realizarse antes de las pruebas funcionales.

Tabla 10 Tipos de pruebas del cuadrante 4

Tipos de Prueba	Descripción	Aplica equipo
Pruebas de escalabilidad	Comprobando que la funcionalidad siga siendo correcta cuando se incrementen más usuarios. Se piensa en todo no solo el software sino también en el hardware correcto que la soporte	NO
Pruebas de performance y carga	Ayudan a identificar los cuellos de botella de la aplicación y su comportamiento con gran cantidad de datos	SI

Pruebas de seguridad	Son las pruebas de seguridad de la aplicación considerando temas de confirmación de identidad, autenticación correcta, autorizaciones de usuario, entre otras	NO
Manejo de memoria	Pruebas verificando el uso de memoria RAM, ROM, acceso de discos duros	NO

Como hemos visto los cuadrantes de pruebas nos ayudan a tener una visión global de que pruebas puedes aplicar en un proyecto, y comparando con lo que realiza el equipo de pruebas de la empresa XYZ vemos que la mayoría no son consideradas.

Estrategias de pruebas

Hay que tener una visión de global de las pruebas que se van a realizar en el contexto de un proyecto, este contexto por lo general se pierde durante las iteraciones entre codificación y pruebas. En la empresa XYZ ocurre lo mismo, se planifica las pruebas o estrategias de pruebas a aplicar y luego se pierde en los detalles.

Un factor de éxito de agile testing es “No se olvide del panorama general” y para ello hay que tener en cuenta los siguientes elementos que confirman el contexto que son: El equipo, el producto y el nivel de detalle del sistema además se debe planificar estrategias de acuerdo con todas las etapas de pruebas ágiles dentro de un proyecto, como se muestra en la figura 4.4 existen 4 etapas.

Estrategia de pruebas ágiles

El ciclo de vida de las pruebas ágiles abarca cuatro etapas

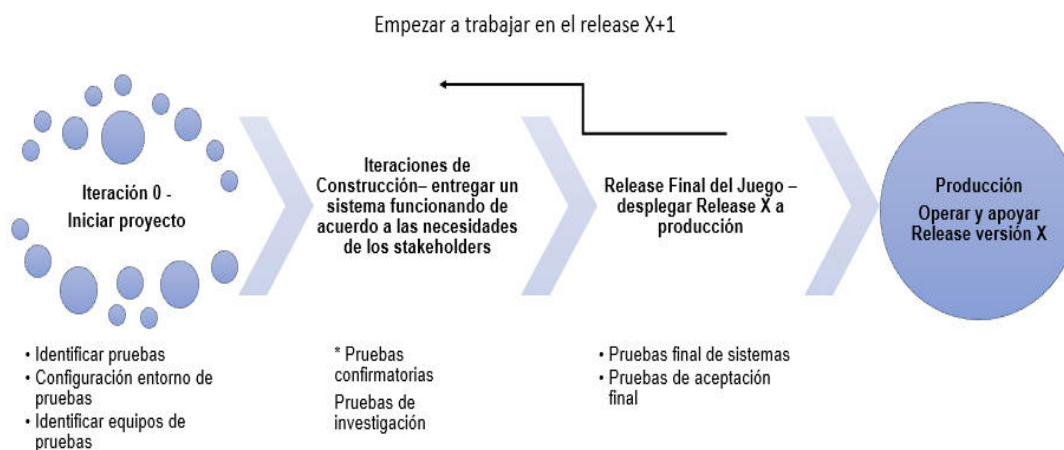


Figura 4.4 Estrategias de pruebas ágiles

En la figura se muestra el ciclo de vida ágil con respecto a las pruebas. La iteración 0 es donde se debe pensar los lineamientos bases para el proyecto, lo que necesitamos como recursos o entorno de pruebas; en la fase de construcción es donde se realizan las diferentes iteraciones y construcción del sistema; en la fase de fin de juego es la transición del nuestro sistema a producción y la fase de producción es la de operación del sistema y apoyo al usuario.

Lo importante de las estrategias de pruebas es que deben realizarse en cada una de estas etapas, ir redefiniendo las estrategias en casos que sean necesarios sin olvidar el panorama macro del proyecto. A continuación, se detalla, las estrategias que se deben considerar en cada una de las etapas.

Tabla 11 Estrategias por Etapas

Etapa	Estrategias	Aplica Equipo
Iteración 0	Se establece los casos de negocio a cubrir en el proyecto	SI
	Preparación del entorno del proyecto y ambientes de pruebas	SI
	Definición de los requerimientos críticos	NO
	Conocer la arquitectura de la aplicación	NO
	Definir los riesgos de pruebas	NO
	Contemplar capacitaciones necesarias	NO
Iteraciones de Construcción	Considerar las pruebas de confirmación que se van a realizar	SI
	Pruebas de investigación que ayuden a determinar problemas potenciales	NO
	Considerar como y cuando realizar las pruebas de regresión	NO
	Cómo y cuándo realizar las pruebas de aceptación	NO
	Se va a realizar pruebas unitarias	NO
Release End Game	Formación a los usuarios	NO
	Personal de apoyo a los usuarios	SI
	Pruebas completas del sistema	SI
	Pruebas de aceptación	NO
Producción	Puesta en producción	NO
	Apoyo a producción	SI

Automatización de pruebas

Otra de las prácticas principales del Agile testing es la automatización de las pruebas con el propósito que sean realizadas más rápida que realizarlas manualmente. [8]

Agile Testing indica que debería de automatizarse lo más que se pueda, de acuerdo a la pirámide de automatización que se muestra en la figura 4.5, donde se muestran 3 niveles de automatización; el nivel más bajo soporta a los demás son la automatización de las pruebas unitarias, es preferible tener toda la codificación con pruebas unitarias automáticas. El nivel intermedio, corresponde a la automatización de las pruebas funcionales, las pruebas de aceptación, las pruebas de Api, más orientado negocio y el funcionamiento por ejemplo reglas de negocios que no cambian. El nivel más alto son las pruebas de UI que indica que debe ponerse el menor esfuerzo, estas son las pruebas orientadas al UI de la aplicación, por lo general son los que más cambian y es por eso que no se debe poner mucho esfuerzo en realizarlas. No importa cuán automatizado se encuentre un proceso, siempre hay una parte de pruebas manuales que pueden realizarse como pruebas exploratorias y las pruebas de aceptación ya que los clientes son los que realizan el uso del sistema de manera manual.

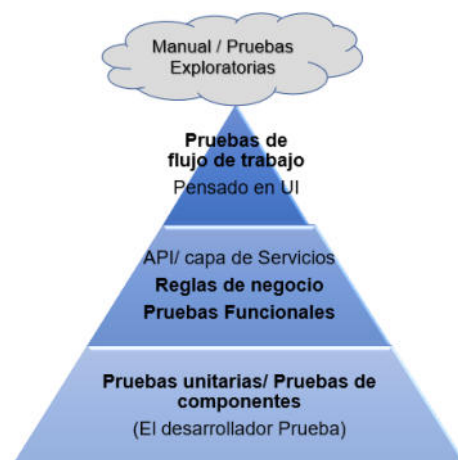


Figura 4.5 Pirámide de automatización de pruebas

La empresa XYZ y su equipo de pruebas ha desarrollado un proyecto de automatización de una aplicación que casi no cambia en su funcionamiento porque fue desarrollada orientada al UI y cuando existe un cambio, se ejecutan las pruebas automáticas de dicha aplicación. Las demás aplicaciones se realizan pruebas manuales, lo que ocasiona que el tiempo de pruebas se incremente durante las ejecuciones.

Como podemos ver en la tabla se compara la automatización de pruebas indicada por Agile Testing con lo que aplica el equipo de pruebas y su impacto.

Tabla 12 Relación pirámide de automatización con el equipo

Pirámide de pruebas	Aplica Equipo
Automatización de pruebas unitarias	NO
Automatización de funcionalidad	NO
Automatización de UI	NO
Pruebas Exploratorias	NO

Proceso de trabajo de Agile Testing

Agile testing propone la realización de las actividades de pruebas dentro de un proyecto ágil como se muestra en la figura 4.6. Como ya lo hemos revisado, el proceso de pruebas inicia desde la concepción de un proyecto, cambio o nuevos requerimientos. Es parte de cada uno de los procesos dentro del marco de trabajo en Scrum. [7]

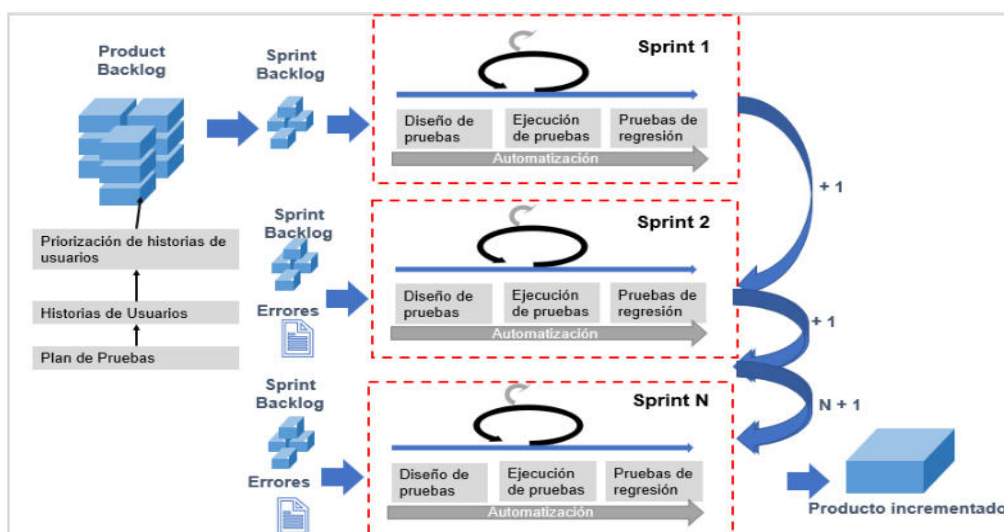


Figura 4.6 Principales actividades del Agile Tester

Podemos ver el inicio con el plan general de pruebas, en donde se busca las necesidades y estrategias de pruebas que se necesitan aplicar de acuerdo al proyecto que se está presentado, es parte de la definición y priorización del Product Backlog, ayuda al equipo en la definición de las historias de usuario, luego es parte de la reunión del sprint también con el punto de vista de calidad e inyectar calidad al proyecto, definiendo tiempos y principalmente aportando con estrategias de pruebas a ser aplicadas dentro del sprint.

Ya dentro del cada sprint, las actividades son las mismas, se realiza el diseño de pruebas, la ejecución de las pruebas, el reporte de defectos, las pruebas de regresión, y en todas las actividades de relación con el equipo cliente y con el equipo de desarrollo, va de la mano la automatización en cada iteración que se está implementando.

Las actividades dentro de la entrega del producto también son consideradas como parte las actividades a realizar por el equipo de pruebas.

En la siguiente tabla, se indica los hitos macros del proceso sugerido por Agile Testing, lo que no realiza el equipo de pruebas de la empresa XYZ y su impacto.

Tabla 13 Proceso Agile Testing macro

Hito	Aplica Equipo
Plan de pruebas	No
Historias de usuario	Si
Estrategia de pruebas del release	No
Estrategias de pruebas del sprint	No
Diseño de pruebas	Si
Ejecución	SI
Pruebas de regresión	Si
Pruebas de Integración	Si
Pruebas de aceptación	No

4.2 ACTIVIDADES DE AGILE TESTING EN LA REUNIÓN DE PLANIFICACIÓN DEL RELEASE

Estas actividades están centradas en tener la visión global del proyecto, entender cuáles son las necesidades del cliente, cuáles son los objetivos en un nivel alto. De acuerdo con esta visión global podemos empezar a planificar qué estrategias de pruebas se deben emplear, qué recursos son necesarios para la realización del proyecto, cuánto nos tomaría las tareas de pruebas de acuerdo con la historia, qué necesidades de automatización vamos a realizar y todo lo que se necesita para llevar el proyecto con la satisfacción que requiere el cliente. Todo en una visión a alto nivel.

A continuación, se detalla las actividades del Agile Tester en la planificación del release.

Midiendo las Historias de usuario

Todo el equipo realiza la medición de la historia de usuario, algunos solo indicando una medición como alto, medio, bajo, otros usan la técnica del planning poker para definir el tiempo que tomaría liberar una historia de usuario. Debe ser considerada el tiempo de las pruebas y es aquí donde interviene la experiencia del Tester.

Priorización de Historias de usuario

El Tester puede ayudar a decidir cual o cuales son las historias de usuarios principales que dan valor al cliente y que se pueden probar.

Medir el impacto en todo el Sistema

El trabajo del agile tester es tener siempre presente, de mirar todo el panorama general y debe pensar en cómo cada historia afecta en conjunto al sistema.

Realizar plan de pruebas

Una vez que se tiene claro y de manera global lo que necesita el cliente, es necesario realizar un plan de pruebas que permita disminuir los riesgos que se presenten en el proyecto desde el punto de vista de pruebas. Se debe considerar las estrategias de automatización, los ambientes de pruebas que serán necesarios, los datos de pruebas que se necesitarán, las pruebas que serán necesarias de acuerdo con la estrategia de los cuadrantes de pruebas. Lo principal es ver todas esas necesidades, plasmarlas y llevar a cabo el plan propuesto.

Este plan de pruebas puede ser tan detallado como sea necesario o puede ser muy simple, lo ideal es plasmar las necesidades presentadas en el release.

Otra manera de realizar un plan sin mucho detalle es usar la Matriz de Pruebas, que es una forma sencilla de comunicar la funcionalidad a probar, esta brinda al equipo una visión rápida de las pruebas necesarias.

Las hojas de cálculo de pruebas pueden ayudar a poner las funcionalidades a ser afectadas por las historias de usuario, y todo el equipo enlistar donde se relacionan, en otra hoja pueden ponerse los riesgos, entre otras

consideraciones que ayudan al equipo de pruebas a realizar una estrategia de pruebas de acuerdo con lo indicado.

Un pizarrón, si el equipo es pequeño o el reléase es pequeño mucha documentación puede no ser necesaria, lo ideal es que se diagrame o dibuje en una pizarra los riesgos o necesidades que pueda traer ese proyecto y quedan visibles para todo el equipo.

En la siguiente tabla se detalla las actividades que realiza actualmente el equipo y su impacto.

Tabla 14 Actividades dentro de la reunión del release

Actividad	Aplica Equipo
Midiendo las Historias de usuario	Si
Priorización de Historias de usuario	Si
Medir el impacto en todo el Sistema	No
Realizar plan de pruebas	No

4.3 ACTIVIDADES DE AGILE TESTING DURANTE LA PLANIFICACIÓN DEL SPRINT

En la planificación del Sprint todo el equipo piensa en lo que viene en la iteración, qué lecciones aprendidas se deben aplicar, qué ha enseñado la retrospectiva de una iteración anterior, se discuten todas las historias de usuarios de la iteración, se proyectan las tareas que se van a realizar. La reunión de planificación del release debe ser realizada por todo el equipo y, el

agile tester como otro miembro del equipo, debe estar atento de los desvíos del alcance a otras direcciones.

Considerando todos los puntos de vista

El tester debe ponerse en los diferentes puntos de vista, es importante ver todos los aspectos de la historia de usuario como cliente, como desarrollador, como product owner, como todos los involucrados en la creación y uso de la funcionalidad a hora por cada historia de usuario y a un nivel detallado. Escribir las pruebas de aceptación en grupo, es una buena manera de que todos comprendan cómo se va a revisar la aceptación de la historia.

Escribir tarjetas de tareas unificadas.

Como la idea de ágil es codificar con pruebas, las tareas de codificación y pruebas deben ser escritas al mismo tiempo. La idea es que todos los miembros del equipo vean las tareas y estén de acuerdo con ellas preferiblemente todas las tarjetas de la iteración y si no es posible la mayoría. El tester certifica que las tareas de desarrollo incluyan el tiempo de programación de pruebas unitarias y también se encarga de crear tareas de pruebas end to end. También se crean las tareas para obtención de datos de pruebas.

Historias de usuarios Comprobables

Siempre que se inicia a hablar de implementación de una historia, el tester debe pensar en cómo la va a probar, este es el último punto donde puede pensar cómo va a probar esa historia.

Colaborar con clientes

Trabajar en estrecha relación con los clientes o sus representantes como el analista funcional o analista de negocio, es una de las principales actividades del agile tester. Es en este momento que se realizan todas las actividades de pedir ejemplos para completar las historias de usuario.

Hay que realizar preguntas sobre la funcionalidad, comprender las necesidades, hay que obtener formación del negocio y el funcionamiento. Esta será la base para ayudar al desarrollador a impulsar la codificación.

Pruebas y Ejemplos de Alto Nivel

Orientar al equipo de cliente a la realización de ejemplos para poderlos convertir en pruebas, de esta manera se puede ayudar al equipo a comenzar en la dirección correcta. El tester debe pedir estos ejemplos en el detalle de los criterios de aceptación para conocer así el propósito principal de la historia.

Los Mockups pueden transmitir claramente los requerimientos de UI que se necesite, de manera rápida y visual, puede ser en una pizarra, en una hoja o una herramienta colaborativa que permite transmitir visualmente dichos ejemplos.

Documentación de casos de prueba en Alto nivel

Estos casos de pruebas se van presentando a medida que se analiza la historia de usuario y los criterios de aceptación, se pueden ir preparando o escribiendo para no perder la idea. Documentando las estrategias de pruebas para la

iteración que se va a realizar con un poco más de detalle de acuerdo con las historias que contenga la iteración.

La siguiente tabla detalla las actividades de esta sección, lo que realiza el equipo de pruebas de la empresa XYZ y el impacto de su implementación si es el caso de realizarla

Tabla 15 Actividades en la planificación del sprint

Actividad	Aplica Equipo
Considerando todos los puntos de vista	No
Escribir tarjetas de tareas unificadas	No
Historias de usuarios Comprobables	No
Colaborar con clientes	No
Pruebas y Ejemplos de Alto Nivel	No
Documentación de casos de prueba en Alto nivel	Si

4.4 ACTIVIDADES DE AGILE TESTING DENTRO DEL SPRINT

A diferencia de las actividades anteriores, estas actividades se desarrollan dentro de la iteración. En el caso de la empresa XYZ la iteración tiene una duración de 4 semanas.

Diseño de Casos de pruebas

Los primeros días, son para detallar los casos pruebas de las historias de usuario que conforman el sprint es donde más colaborativamente se debe trabajar. También es tiempo de comenzar con el diseño de pruebas

automatizadas, empezando con lo más simple, por la ruta feliz. Si el tiempo para la automatización incurre en más tiempo, es conveniente realizar casos de pruebas manuales.

Como tester se debe estar pensando en nuevos escenarios de pruebas, para pruebas exploratorias o manuales. El propósito es dar ejemplos o escenarios de pruebas a los programadores mientras escriben el código.

En la iteración, desarrollo y pruebas van de la mano. Desarrolladores, testers, expertos en base datos o otros miembros del equipo, trabajan colaborativamente para desarrollar la historia.

Pruebas criticando al producto

Una vez que se encuentre desarrollado y probado los componentes, hay q comprobar orientado al negocio y es donde se comienzan a aplicar las pruebas funcionales. Probar los diferentes escenarios comprando que la historia cumpla con lo que se está solicitando. Una historia no se finaliza sino se han realizado todos los tipos de pruebas planificados en las estrategias.

Existen varias prácticas que permiten trabajar colaborativamente mejorando el producto tal como se indica a continuación.

Pruebas de pares

Las pruebas de pares son realizadas por el equipo, el desarrollador y el tester se sientan juntos revisando que lo codificado funcione de acuerdo a lo esperado, sino no está correcto, ahí mismo es solucionado y probado nuevamente.

“Muéstrame”

Mostrarle a alguien un problema es mucho mejor que tratar de solucionarlo solo, es más efectivo. Si el equipo se encuentra en diferentes horarios, se tiene que buscar al más cercano.

Hablar con clientes

Es preferible tener los casos de pruebas revisados con los clientes o con un representante del negocio, pero si no es así, y existen dudas es preferible preguntar antes que se asuma. Recordando que la comunicación no debe ser técnica, sino usando todas las herramientas de visualización y ejemplos para obtener la información.

Si se encuentra codificado se puede mostrar el avance para que tenga una mejor visión de la pregunta y las repuestas sobre el caso presentado. Es importante llevar el control del cambio y la expectativa sabiendo manejar lo que se pueda realizar en el sprint y lo que se pueda dejar para el siguiente.

Cumplimiento a las tareas de pruebas

En algunas ocasiones se ha indicado que el agile tester es proactivo, no está esperando que le llegue el trabajo. Mientras existe algún tiempo, el agile tester se encuentra haciendo automatización de pruebas UI, así como pruebas de integración o creando nuevos casos manuales de ser necesarios.

Errores

Como hemos visto todas las actividades son realizadas para prevenir errores, más en el caso que existan estos deben ser tratados de la siguiente manera:

- Indicarlo inmediatamente al desarrollador para que se lo solucione inmediatamente.
- En el caso que no se pueda resolver inmediatamente, se agrega como una nueva tarjeta de trabajo.

En el caso de encontrar un error en las pruebas exploratorias, este debe ser documentado y solucionado, pero así mismo se debe automatizar y considerar el caso de prueba.

Otra manera de pensar en errores es los llamado “Deuda Técnica”, que es un error complejo técnicamente de solucionar y se deja para una siguiente iteración porque requiere de cierta investigación. Es preferible que desaparezca lo más pronto posible.

Pruebas de Regresión

Cuando estemos en una iteración +1 es necesario realizar las pruebas de la iteración anterior más las de la iteración actual. Si tenemos automatizado las pruebas de la iteración anterior sería mucho más rápida la ejecución y se continuaría con las tareas de la iteración.

Los desarrolladores corren todas las pruebas unitarias automáticas luego de agregar los cambios de la iteración, si existe un error, este es solucionado inmediatamente y codificado y automatizado para las siguientes iteraciones.

Los testers durante la iteración automatizan las pruebas de la iteración para luego usarla en las siguientes iteraciones, estas deben ser incluidas en las suite de regresión y con su respectiva versión.

Recordemos que no se debe perder el punto de vista general, las pruebas se completan cuando todas las regresiones pasan y no solo las funcionales, las no funcionales que se encuentren en la regresión también deben pasar para dar por terminada la historia.

Si no hay pruebas automáticas de regresión, se debe terminar luego de culminar las pruebas de regresión manuales, sin dejar de hacerlas.

Recursos

Antes de iniciar una iteración debemos tener listo los ambientes de pruebas, los datos y las herramientas para las historias de la iteración de acuerdo con las estrategias de pruebas planificadas antes del inicio del sprint.

Pero si existen necesidades adicionales, se pide colaboración o todos los miembros del equipo.

MÉTRICAS

Durante la iteración es importante ir midiendo las actividades de desarrollo y pruebas y sus avances. Se va midiendo si todo lo realizado ha sido realmente producido con calidad como lo hemos deseado y lo que no se mida no se mejora. Las principales métricas las indicamos a continuación.

Medir el progreso

En cualquier punto de la iteración es importante conocer cómo se va avanzando y lo que queda por realizar lo vamos a terminar en lo planificado. El gráfico burndown mide el progreso del equipo. Se obtiene de las historias completas, que incluye la finalización de las pruebas entre las todas las tareas de la iteración.

Métrica de defectos

Esta métrica no es para saber la cantidad de defectos que se ha encontrado, sería innecesaria esta métrica en Agile Testing. Pero se la puede utilizar durante el proceso de transición de pruebas tradicionales a prueba ágiles.

Es importante conocer en esta métrica, cuál es el origen del error, para saber dónde mejorar.

En la siguiente tabla se indica las actividades dentro del sprint y si las realiza el equipo con su respectivo impacto.

Tabla 16 Actividades dentro del sprint

Actividad	Aplica Equipo
Diseño de casos de pruebas	No
Pruebas criticando al producto	Si
Hablar con clientes	No
Cumplimiento a las tareas de pruebas	Si
Errores	Si
Pruebas de regresión	Si
Recursos	No

Métricas	No
----------	----

4.5 ACTIVIDADES DE AGILE TESTING EN LA LIBERACIÓN

Luego de cada sprint, se realiza la retrospectiva con todo el equipo, el Tester debe proponer ideas de cómo mejorar, en distintos ámbitos, para incluirlas en las siguientes iteraciones.

De acuerdo con scrum existe la reunión “Sprint Review” donde el usuario valida que los criterios de aceptación y comprueba el producto entregado. Si bien esta reunión es un par de horas, donde se podría realizar las pruebas de aceptación del cliente. Siendo cada iteración una parte de la funcionalidad.

Agile Testing propone que luego de las pruebas de cada iteración, se debe estar seguro de que cada uno de esos entregables funcionan juntos apropiadamente. Para esto, se debe planificar como parte de las estrategias desde el inicio del proyecto o pedir tiempo extra para estas pruebas en el caso de ser necesario, pero no deberían dejar de hacerse.

End Game

La iteración “End Game” es el tiempo en el que el equipo hace los últimos arreglos al producto. No se trata de un ciclo de arreglo de errores sino, que corresponde a las verificaciones finales del producto total.

En este paso, el agile tester realizar pruebas exploratorias finales, pero mirando toda la aplicación, es decir, el proceso de negocio de inicio a fin. Certificando que funcione correctamente.

La aplicación debe ubicarse en la ambiente pre-producción muy parecido al de producción. El tester certificará que la puesta a producción se realice satisfactoriamente.

Se realizan pruebas no funcionales finales. Al estar en un ambiente muy parecido a producción se certificará que la aplicación corra con el performance y carga correcta. Así también, queo las configuraciones de los servidores se encuentren de acuerdo con las necesidades de la aplicación.

Si existen aplicaciones externas, se certifica que queden integradas correctamente.

Pruebas de Aceptación

Las pruebas de aceptación corresponden a las pruebas del flujo de negocio luego de culminar el desarrollo del sprint, ésta debe de realizarse por reléase sin dejar para antes de la salida del proyecto.

Estas pruebas son realizadas por el cliente con el apoyo del tester de ser necesario. No es darle los casos de pruebas, sino que el cliente pueda aplicar su experiencia en el negocio y las actividades que realiza el su día a día.

A continuación, tabla de las actividades de liberación, las que aplica el equipo y su impacto en la implementación de esa actividad.

Tabla 17 Actividades de liberación del sprint

Actividades	Aplica Equipo
Retrospectiva	Si
Sprint Review	Si
End Game	No
Pruebas de Aceptación	No

4.6 PRÁCTICAS AGILE TESTING COMPRANDO EL PROCESO ACTUAL

Una vez analizada las estrategias que propone Agile Testing, las comparamos con el subproceso de pruebas para tener claro las estrategias que de una u otra manera se realizan y otras que no.

Con respecto a las Estrategias de pruebas

Vemos en la siguiente tabla la comparación de las estrategias de los cuadrantes de Agile Testing con relación al proceso actual.

Tabla 18 Comparación Cuadrantes con Proceso pruebas actual
Estrategia Cuadrante de Pruebas

Tipos de Prueba	Actividad del Proceso Actual
Pruebas unitarias	
TDD - test driven development (Desarrollo dirigido por pruebas)	
Pruebas de componentes	
Pruebas unitarias Manuales	
Herramientas para levantamiento de Ejemplos y requerimientos	
Pruebas funcionales Manuales	Ejecutar Pruebas
Pruebas funcionales automatizadas	
Pruebas de regresión	Ejecutar Pruebas
Pruebas de regresión automática	

Pruebas de prototipo de interfaz con el usuario	
Pruebas Exploratorias	
Pruebas de usabilidad	
Pruebas de Aceptación	
Pruebas de escalabilidad	
Pruebas de Performance y Carga	Ejecutar Pruebas
Pruebas de Seguridad	
Manejo de memoria	

Como vemos las mayorías de las estrategias con respecto a las cuadrantes no son realizadas.

La siguiente tabla compara las Estrategias al proceso con Agile Testing por etapas de desarrollo con relación a la actividad donde se realiza o no en el proceso actual.

Tabla 19 Comparación Estrategias del proceso contra el proceso actual de pruebas

Etapas	Estrategias	Actividades proceso Actual
Iteración 0	Se establece los casos de negocio a cubrir en el proyecto	Definir historias de usuarios a cubrir
	Preparación del entorno del proyecto y ambientes de pruebas	Preparar ambientes de pruebas
	Definición de los requerimientos críticos	
	Conocer la arquitectura de la aplicación	
	Definir los riesgos de pruebas	
	Contemplar capacitaciones necesarias	
	Considerar las pruebas de confirmación que se van a realizar	Documentar casos de pruebas

Iteraciones de Construcción	Pruebas de investigación que ayuden a determinar problemas potenciales	
	Considerar como y cuando realizar las pruebas de regresión	
	Cómo y cuándo realizar las pruebas de aceptación	
	Se va a realizar pruebas unitarias	
Release End Game	Formación a los usuarios	
	Personal de apoyo a los usuarios	
	Pruebas completas del sistema	
	Pruebas de aceptación	
Producción	Puesta en producción	Liberar Release
	Apoyo a producción	

Estrategia de automatización de pruebas

Con respecto a la estrategia de automatización de pruebas, no se puede comprobar con el proceso actual porque actualmente no se realiza automatización de pruebas. En la siguiente tabla muestra el impacto de su implementación.

Comparación con las Actividades de planificación

Vamos a comparar las actividades que propone agile Testing con respecto a las planificaciones del proyecto y del sprint.

Tabla 20 Comparación Actividades de planificación de Agile Testing y proceso actual

Actividad	Actividades proceso actual
Midiendo las Historias de usuario	Medir historias de usuarios
Priorización de Historias de usuario	Priorizar historias de usuarios
Medir el impacto en todo el Sistema	
Realizar plan de pruebas	
Considerando todos los puntos de vista	

Escribir tarjetas de tareas unificadas	
Historias de usuarios Comprobables	
Colaborar con clientes	
Pruebas y Ejemplos de Alto Nivel	
Documentación de casos de prueba en Alto nivel	Documentar casos de pruebas

Actividades de Dentro del Sprint

En la siguiente tabla se compara las actividades de pruebas dentro del sprint, es decir, la realización del sprint.

Tabla 21 Comparación actividades dentro del Sprint con actividad del proceso actual

Actividad	Actividad del proceso Actual
Diseño de casos de pruebas	Documentar casos de pruebas
Pruebas criticando al producto	Ejecutar de pruebas
Hablar con clientes	
Cumplimiento a las tareas de pruebas	
Colaborar con desarrollo	
Errores	Reportar errores
Pruebas de regresión	Ejecutar pruebas
Recursos	
Métricas	

Comparación de actividades en la liberación

En la siguiente tabla comparamos las actividades finales luego del sprint, comparadas con el proceso actual.

Tabla 22 Comparación actividades liberación con proceso actual

Actividades	Actividad el proceso actual
Retrospectiva	Realizar retrospectiva
Sprint Review	
End Game	Realizar revisión
Pruebas de Aceptación	

Con la comparación de todas las propuestas de agile Testing con el proceso actual de pruebas, hemos comprobado las mejoras que podemos realizar al actual proceso considerando no solo actividades como tal, sino también estrategias que se pueden transformar en actividades.

4.7 PRÁCTICAS DE AGILE TESTING PROPUESTOS PARA SOLUCIONAR PROBLEMAS DEL PROCESO ACTUAL












Como hemos revisado en las secciones anteriores de este capítulo, las prácticas que hemos revisado de Agile Testing nos permiten mejorar el proceso actual. En esta sección vamos a revisar cuáles de dichas actividades y prácticas permiten solucionar los problemas críticos que se han evaluado del proceso actual y su impacto, principalmente en el equipo de pruebas.

Aunque cada una de las prácticas, nos ayudan a mejorar la calidad del producto, también pueden causar un impacto positivo o negativo al incluirlo en la mejora del proceso. En la columna Impacto, se indica el efecto que causará la introducción de la actividad o estrategia como propuesta de mejora para el subproceso de pruebas. El color verde indica que el impacto es positivo para el proceso y naranja causa un impacto negativo.

También se ha incluido la columna seleccionar, la misma que indica una propuesta de cuál de las prácticas de agile Testing pueden ayudar a mejorar el Subproceso de pruebas.














Con respecto a los problemas críticos de la etapa de Planificación y estimación, veamos la siguiente tabla.










Tabla 23 Prácticas de Agile Testing a problemas críticos de etapa de planificación

Problemas Crítico	Práctica Agile Testing	Impacto	Seleccionar	
No se incluye a todos los miembros del equipo de pruebas para el diseño del producto.	La calidad es responsabilidad de todo el equipo.		Disminuye los errores en producción.	<input checked="" type="checkbox"/>
	Considerando todos los puntos de vista de cada equipo.		Disminuye los errores durante el sprint	<input checked="" type="checkbox"/>
	Colaborar con el equipo cliente		Disminuye los errores en producción	<input checked="" type="checkbox"/>
No se diseñan las estrategias de pruebas en conjunto entre los equipos de desarrollo y pruebas	Plan de pruebas visual		Disminución los errores durante el sprint	
	Realizar pruebas y ejemplos de Alto nivel		Disminución los errores durante el sprint	<input checked="" type="checkbox"/>
	Documentación de pruebas en Alto nivel		Disminución los errores durante el sprint	<input checked="" type="checkbox"/>
	Los casos de pruebas deben ser compartidos con los desarrolladores		Disminución los errores durante el sprint	<input checked="" type="checkbox"/>
Falta de comunicación con el equipo cliente	El PO indica todas las pruebas de negocio que son necesarias realizar		Disminuye los errores en producción.	
	El Agile Tester debe estar pendiente de saber todas las expectativas del cliente.		Disminuye los errores en producción.	
	Colaborar con clientes		Disminuye los errores en producción.	<input checked="" type="checkbox"/>
	Obtener todos los ejemplos de negocio posible		Disminuye los errores en producción.	<input checked="" type="checkbox"/>

Con respecto a los problemas críticos en la etapa de implementación



Tabla 24 Practicas de Agile Testing a problemas críticos de implementación





Problemas Crítico	Práctica Agile Testing	Impacto	Seleccionar
No se realizan pruebas unitarias en el desarrollo.	La calidad es responsabilidad de todo el equipo.	 Disminuye los errores en producción.	<input checked="" type="checkbox"/>
	Los casos de pruebas deben ser compartido con los desarrolladores	 Disminuye los errores dentro del sprint.	<input checked="" type="checkbox"/>
	Realizar pruebas unitarias automáticas	 Disminuye errores en producción.	<input checked="" type="checkbox"/>
		 Decrementa productividad.	
Se da paso al proceso a pruebas solo cuando el producto está terminado	Trabajar colaborativamente en pruebas antes de dar el paso a verificaciones.	 Decrementa productividad  Disminuye errores dentro del sprint	<input checked="" type="checkbox"/>
	Realizar técnica de pruebas de pares	 Disminuye los errores dentro del sprint.	
	Técnica "Muéstrame"	 Disminuye los errores dentro del sprint.	
Poco tiempo disponible para realizar las pruebas antes de que termine el sprint	Los errores reportarlos y solucionarlos inmediatamente	 Mejora la productividad.	<input checked="" type="checkbox"/>
	Solo se quedan los errores en el backlog	 Incremento de errores en el Product backlog.	
	Realizar técnica de pruebas de pares	 Disminuye los errores dentro del sprint.	<input checked="" type="checkbox"/>
	Automatización de casos de pruebas y UI	 Disminución de la productividad.	
	Pruebas Exploratorias	 Incrementa productividad	<input checked="" type="checkbox"/>

			en ejecución de pruebas.	
No existe automatización de casos de pruebas	Pruebas de regresión automatizadas		Disminución de la productividad.	
	Automatización de pruebas unitarias	 	Disminución de la productividad. Disminuye errores en producción	<input checked="" type="checkbox"/>
	Automatización de UI		Disminución de la productividad.	
Existe gran cantidad de errores durante las pruebas	Realizar técnica de pruebas de pares		Disminuye los errores dentro del sprint.	<input checked="" type="checkbox"/>
	Técnica "Muéstrame"		Disminuye los errores dentro del sprint.	
	Indicar a desarrollo todos los casos de pruebas a realizar.		Disminuye los errores dentro del sprint.	<input checked="" type="checkbox"/>
	Trabajar en prevenir errores.		Disminuye los errores dentro del sprint.	<input checked="" type="checkbox"/>
	El PO debe estar pendiente de cualquier inquietud sobre expectativas del cliente		Disminuye los errores dentro del sprint.	<input checked="" type="checkbox"/>

Con respecto a los problemas críticos en la revisión del sprint

Tabla 25 Prácticas de Agile Testing a problemas críticos en revisión

Problemas Crítico	Práctica Agile Testing	Impacto	Seleccionar
No se realizan pruebas de aceptación.	Práctica del End game.		Disminuye los errores en producción.
	Realizar pruebas de aceptación		Disminuye los errores en producción.

	El PO está pendiente de las pruebas de casos de negocio		Disminuye los errores en producción.	
No se realiza un proceso de puesta en producción	Puesta a producción		Disminuye los errores en producción.	
	Apoyo a producción		Disminuye los errores en producción.	

Este análisis nos ha permitido darnos cuenta cuáles de las prácticas de agile Testing pueden ayudar a la solución de los problemas más críticos que ha presentado el equipo y de ellos proponer algunos para ser parte del rediseño del proceso.

CAPÍTULO 5

5 PROPUESTA DE MEJORA DEL PROCESO

5.1 REDISEÑO DEL PROCESO

Para rediseñar el subproceso de pruebas dentro del proceso de desarrollo de software de la empresa empezamos por definir cuáles son los objetivos que se busca alcanzar con este rediseño y estos son:

Tabla 26 Objetivos proceso

Objetivo	Métrica
Incrementar el desarrollo de SCRUM en la empresa.	Resultado de autoevaluación de la organización de nivel de Scrum
Mejora de la satisfacción de los clientes por medio de la reducción del número de defectos críticos generados por sprint.	Medición de defectos críticos reportados por clientes correspondientes a un Sprint.

Mejora de la eficiencia del desarrollo del software.	Medición de puntos desarrollados por sprint mediante el sistema establecido por la organización.
--	--

Uno de los problemas más grandes que ha tenido la organización es la falta de integración y poca colaboración entre equipos. En el diagrama As-Is tal como mostramos en capítulos anteriores, se nota que las actividades son realizadas en su mayoría por el equipo de pruebas manteniendo el mínimo de comunicación con otros roles.

Uno de los cambios más importantes es la inclusión de los diferentes equipos involucrados en el subproceso con lo que tenemos los siguientes roles definidos:

Tabla 27 Roles y actores

Rol	Descripción	Actor
Product Owner	Responsable de todo el proyecto.	Mr. George Harris
Equipo de pruebas	Equipo responsable del aseguramiento de calidad del producto.	Angela Castillo, Carolina Buenaño, Alexandra Perez
Equipo de desarrollo	Desarrolladores de software.	Jorge Gamboa, Mariana Gracia, Pedro Rodriguez, Alvaro Jimenez
Equipo cliente	Responsable de la comunicación e interacción con el cliente	Mr. Andrew James

Manteniendo las mismas etapas originales que son Planificación y estimación, Implementación y Revisión del Sprint, nuestra propuesta hace cambios tanto en las actividades como en el aspecto colaborativo de su desarrollo.

Las actividades que han sido incluidas en el subproceso rediseñado parten de la inclusión de buenas prácticas de Agile testing. En la siguiente tabla podemos ver la relación entre estas prácticas y las actividades que se han agregado al subproceso.

Tabla 28 Relación Actividad con Agile Testing

Agile Testing	Actividad relacionada
La calidad es responsabilidad de todo el equipo. Considerando todos los puntos de vista de cada equipo.	Establecer requerimientos
	Medir historias de usuario
	Medir impacto sobre el sistema
	Revisar impacto
	Considerar todos los puntos de vista
Obtener todos los ejemplos de negocio posible. Colaborar con clientes. Realizar pruebas y ejemplos de Alto nivel. Documentación de pruebas en Alto nivel. Los casos de pruebas deben ser compartidos con los desarrolladores.	Generar ejemplos de alto nivel
	Revisar casos de clientes
	Escribir tarjetas de tareas de pruebas
	Escribir tarjetas de tareas de desarrollo
	Unificar tarjetas de tareas
	Establecer plan de pruebas
	Definir estrategias de pruebas
	Crear documentación de alto nivel
La calidad es responsabilidad de todo el equipo. Los casos de pruebas deben ser compartido con los desarrolladores. Realizar pruebas unitarias automáticas. Trabajar colaborativamente en pruebas antes de dar el paso a	Preparar ambientes de prueba
	Revisar estrategias de prueba
	Definir casos de prueba
	Iniciar TDD
	Realizar Pruebas generales

verificaciones. Los errores reportarlos y solucionarlos inmediatamente. Realizar técnica de pruebas de pares. Pruebas Exploratorias. Indicar a desarrollo todos los casos de pruebas a realizar. Trabajar en prevenir errores.	Colaborar con el equipo de desarrollo
	Realizar pruebas unitarias
	Generar retroalimentación
	Revisar retroalimentación
	Generar reporte de defectos
	Calcular métricas de defectos
	Realizar pruebas end to end
Realizar Pruebas de aceptación. El PO debe estar pendiente de cualquier inquietud sobre expectativas del cliente.	Liberar release
	Realizar pruebas de aceptación
	Realizar reunión de retrospectiva
	Realizar revisión final del Sprint

Enfocándonos en el rediseño y cómo cambia el subproceso, utilizamos la siguiente tabla para mostrar una breve descripción de las actividades y su respectivo cambio de existir en el modelo To-Be respecto a la etapa de Planificación y estimación.

Tabla 29 Actividades durante la planificación

As-IS	To-Be	Rol/es
Define los requerimientos como un actor externo.	Define los requerimientos como parte del equipo Scrum	Product Owner
Se realiza una medición de historias de usuario	Se realiza una medición de historias de usuario	Equipo de pruebas
Se priorizan las historias de usuario	Se hace una medición inicial del impacto sobre el sistema	Equipo de pruebas
No existe	Se hace una revisión del impacto incluyendo a varios equipos	Equipo de pruebas, Equipo de desarrollo, Técnicos, Product Owner

No existe	Se consideran todos los puntos de vista respecto a las historias de usuario	Equipo de pruebas, Equipo de desarrollo, Técnicos, Product Owner
No existe	Se generan ejemplos de alto nivel	Equipo cliente
No existe	Se revisan los casos del equipo cliente	Equipo de pruebas
No existe	Se escriben tarjetas de tareas de pruebas	Equipo de pruebas
No existe	Se escriben tarjetas de tareas de desarrollo	Equipo de desarrollo
No existe	Se unifican las tarjetas de tareas	Equipo de pruebas
No existe	Se establece un plan de pruebas	Equipo de pruebas
No existe	Se definen estrategias de pruebas	Equipo de pruebas
Documentar casos de pruebas	Se genera documentación de alto nivel	Equipo de pruebas

En la siguiente tabla podemos ver una breve descripción de las actividades y su respectivo cambio de existir en el modelo To-Be respecto a la etapa de Implementación.

Tabla 30 Actividades durante la implementación

As-IS	To-Be	Rol/es
Se preparan los ambientes de pruebas	Se preparan los ambientes de pruebas	Equipo de pruebas
No existe	Se revisan estrategias de pruebas	Equipo de pruebas
No existe	Se definen casos de pruebas	Equipo de pruebas
Se ejecuta ciclo de desarrollo y pruebas	Se ejecuta desarrollo dirigido por pruebas, el proceso ya no es un ciclo sino un paralelismo entre pruebas y desarrollo	Equipo de pruebas, equipo de desarrollo

No existe	El equipo cliente genera retroalimentación durante la implementación	Equipo cliente
Se generan informes diarios e defectos	Se genera un reporte de defectos	Equipo de pruebas
Se calculan métricas de defectos	Se calculan métricas de defectos	Equipo de pruebas

En la siguiente tabla podemos ver una breve descripción de las actividades y su respectivo cambio de existir en el modelo To-Be respecto a la etapa de Revisión del sprint.

Tabla 31 Actividades durante la Revisión del sprint

As-IS	To-Be	Rol/es
No existe	El equipo de pruebas realiza pruebas finales generales (End to End)	Equipo de pruebas
Se libera el release	Se libera el release	Equipo de pruebas
No existe	Se hace una revisión general de aceptación del producto	Product Owner
Se realizar una reunión para revisar la retrospectiva del Sprint	Se realizar una reunión para revisar la retrospectiva del Sprint	Equipo de pruebas, Equipo de desarrollo, Técnicos, Product Owner
Se realizar una revisión final del trabajo realizado durante el Sprint	Se realizar una revisión final del trabajo realizado durante el Sprint	Equipo de pruebas

5.2 IMPACTO DEL REDISEÑO

Como hemos podido ver en el punto anterior, se ha pensado en agregar tareas que antes el equipo de prueba no ha realizado principalmente porque no

considera aun el trabajo colaborativo con los demás equipos. Esto nos indica que el cambio de las actividades va relacionado directamente con la cultura del equipo en donde vamos a ver el mayor impacto.

Con respecto a los cambios dentro de la planificación hemos analizado como impacta el nuevo proceso en la siguiente tabla.

Tabla 32 Impacto en la planificación

¿Cómo afecta a los clientes?	Con el cliente directamente en caso de ser necesario, se trabajaría con los consultores, que son los que atienden directamente al cliente.
	Hacerlos parte del proyecto permitirá conocer inmediatamente su retroalimentación.
	Permitirá obtener de ellos, ejemplos de casos de negocio que aportarán con características mejor orientadas al usuario final.
¿Cómo afecta a los empleados?	Con respecto al equipo de pruebas, cambiar la manera de pensar individual y pensar de manera colaborativa directamente con el equipo de cliente y con el equipo de desarrollo.
	Se puede ver como las tareas de planificación han crecido, lo que puede ocasionar que el equipo indique que se ha incrementado el trabajo. Pero hay que demostrarles que son tareas de prevención y así evitar que en la etapa de implementación se incremente los tiempos.
¿Cómo impacta en los sistemas?	Van a necesitar mucho acompañamiento en este cambio de mentalidad para todo el equipo. Todos deben pensar que el proyecto con calidad es responsabilidad de todos.
	Actualmente se usa Team Foundation para el control de desarrollo y versión de fuentes de los programas, se debe invertir en licencias para incrementar la versión de pruebas. De esta manera se relacionan los proyectos y las estrategias de pruebas en la misma herramienta.
	Actualizar TFS a versión 2019 para poder utilizar planificación de proyectos ágiles

Con respecto a la implementación del producto, ya dentro del sprint se ha analizado de acuerdo con la siguiente tabla.

Tabla 33 Impacto durante implementación

¿Cómo afecta a los Infraestructura?	Hasta que el equipo se encuentre maduro, con respecto a temas de automatización, el impacto va a ser alto y el avance lento por lo que es recomendable incorporar nuevos roles como tester de automatización.
	Se cambia el líder QA para que realice el rol de SCRUM Master y ayude en los equipos en la liberación de lo planificado.
¿Cómo afecta a los empleados?	Tanto para el equipo de pruebas como para el equipo de desarrollo es necesario capacitaciones de agile testing, Tdd, Atdd, integración continua.
	El aprendizaje de la creación de pruebas unitarias automatizadas y tener los casos de pruebas antes de iniciar
	Es necesario capacitaciones de habilidades blandas como es el caso de mejorar la comunicación entre los equipos.

Con respecto a finalización del Sprint se ha analizado el impacto de acuerdo a la siguiente tabla.

Tabla 34 Impacto finalización

¿Cómo afecta a los clientes?	El realizar las pruebas finales con el cliente o el equipo cliente es un reto de gran impacto. Se debe disponer de los recursos necesarios para estas actividades.
-------------------------------------	--

5.3 PROCESO MACRO DEL PROCESO DE DESARROLLO MEJORADO

En el proceso macro de desarrollo en general puede que no se vea un gran cambio porque se sigue en el proceso del Marco de trabajo SCRUM para el desarrollo de nuevos productos.

Como podemos apreciar en la figura 5.1 el proceso de desarrollo se muestran los mismo equipos desarrollo, pruebas y cliente cada uno desempeñando su respectivo Rol. Para aclarar el rol de los miembros de los equipos a continuación, se muestra la siguiente tabla:

Tabla 35 Roles proceso mejorado

Actual		Cambio
Equipo	Rol	Rol
Desarrollo	Líder de producto	Desarrollador
Desarrollo	Desarrollador junior	Desarrollador
Pruebas	Líder QA	Scrum Master
Pruebas	Analista de Calidad	Agile Tester
Cliente	Consultores	Consultores
Cliente	Product Owner	Product Owner
Cliente	Soporte	Soporte

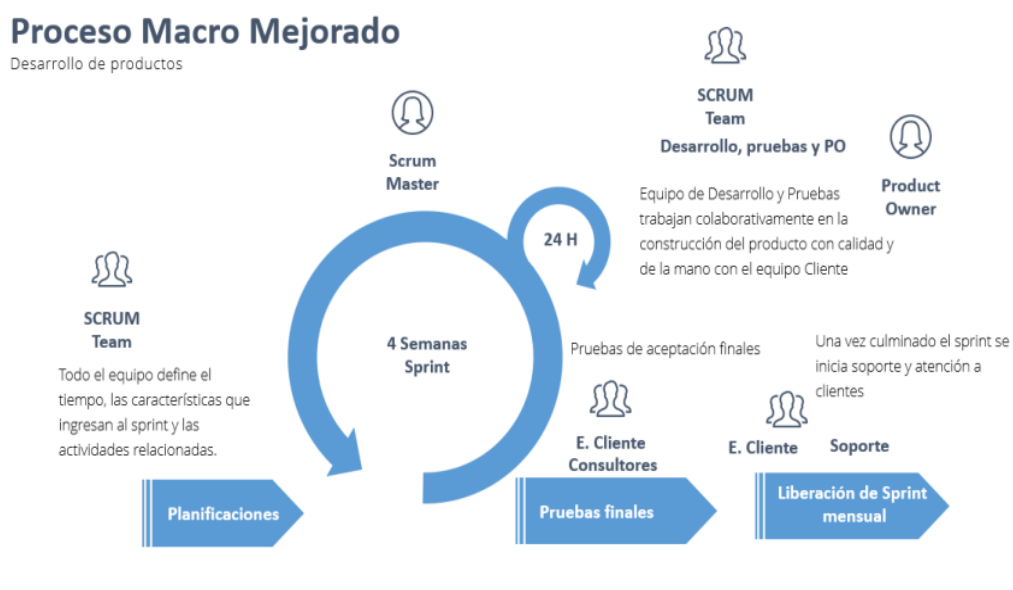


Figura 5.1 Proceso Macro mejorado

El proceso inicia con la planificación del sprint en equipo se decide que características pueden ser incluidas en el sprint, todos los equipos participan de la planificación del sprint, pero ahora se definen tareas colaborativas entre los equipos, las estrategias de pruebas son realizada por todo el equipo, así como la obtención de datos para la certificación de las historias de usuario.

Dentro del sprint, el equipo scrum formado por los desarrollares y los agile testers realizan la implementación y pruebas de acuerdo con la implementación de TDD y automatización de pruebas mientras se realiza la implementación de las nuevas características y van de la mano con el PO para poder comprender las necesidades de los clientes, si es necesario, dentro del Sprint.

Así también hemos incluido el rol del Scrum Master, que actualmente se realiza de manera informal por parte del líder QA, ahora desempeñaría el rol de Scrum master facilitando las diferentes necesidades que se presenten en los equipos.

Una vez concluido todo el proceso de implementación, las pruebas finales las realiza el consultor, que es parte del equipo cliente. Comprobando que todo lo solicitado es entregado de acuerdo con las necesidades del negocio.

Una vez todo concluido es liberado el sprint para luego realizar las respectivas instalaciones o puesta en producción en los clientes que deseen esa versión liberada.

5.4 MODELO TO - BE DEL SUBPROCESO DE PRUEBAS

Siguiendo con el ciclo de BPM en este punto hacemos referencia al rediseño del proceso. El nuevo modelo empieza en la etapa de Planificación y estimación con el Product Owner y la definición de requerimientos para el Sprint. Estos requerimientos pasan al equipo de pruebas quienes se encargan de hacer la medición de las historias de usuarios y de hacer la medición inicial del impacto de lo que se va a agregar o cambiar sobre el sistema. Se realiza una reunión en la que se involucra tanto el Product Owner como los equipos de desarrollo y pruebas y se hace una revisión del impacto en conjunto y se consideran todos los puntos de vista para el desarrollo. Una vez terminado esto, se avanza con el equipo cliente, quienes tienen la tarea de generar ejemplos de alto nivel que faciliten el desarrollo y estos ejemplos son revisados por el equipo de pruebas.

Siguiendo con las actividades, tanto los equipos de pruebas como de desarrollo escriben sus respectivas tarjetas de pruebas y desarrollo, las cuales una vez que están terminadas son unidas en una actividad posterior por el equipo de pruebas.

Para terminar la primera etapa, el equipo de pruebas trabaja en establecer un plan de pruebas, definir las estrategias que se van a seguir para realizar las pruebas y se genera documentación de alto nivel.

La segunda etapa, Implementación, empieza con el equipo de pruebas haciendo la preparación de los ambientes de prueba. Una vez listo esto revisan la estrategia de pruebas y definen los casos de pruebas. En este punto, el equipo de desarrollo está listo para empezar con el desarrollo basado en pruebas TDD y empieza a generar el producto. Aquí es importante recalcar que se hace un trabajo paralelo de desarrollo y pruebas que permite al equipo de pruebas no tener que esperar al final del desarrollo para poder realizar las mismas. En este punto se estrecha la interacción y colaboración del equipo de pruebas con el equipo de desarrollo. Este trabajo paralelo entre desarrollo y pruebas es realmente un ciclo y cada vez que una parte del producto está lista se pasa al equipo cliente para poder tener su retroalimentación. Si el equipo cliente indica que hay cosas que deben cambiarse o no está conforme con lo que evaluaron, el equipo de pruebas revisa esta retroalimentación y vuelve al ciclo de desarrollo.

Una vez que el ciclo anterior concluye, el equipo de pruebas debe generar un reporte de defectos y calcula las métricas relacionadas concluyendo la etapa de implementación.

Finalmente, en la etapa de revisión del Sprint, el equipo de pruebas realiza pruebas generales “end to end” con lo que se da paso a la liberación del release. Habiendo recibido el release, el Product Owner realizar pruebas de aceptación de alto nivel y convoca a los equipos involucrados para una reunión de retrospectiva del Sprint. La etapa termina con el equipo de pruebas realizando una revisión final general del Sprint.

Un modelo BPM que representa lo anteriormente escrito es el que se encuentra en el anexo 2.

CAPÍTULO 6

6 IMPLEMENTACIÓN Y RESULTADOS

6.1 DEFINICIÓN DE LA IMPLEMENTACION

Siguiendo con el ciclo BPM, en este punto hacemos referencia a la implementación del proceso rediseñado.

Para poder ejecutar la implementación del subproceso de pruebas mejorado hemos establecido un plan seccionado en 3 fases tal como se describe a continuación:

Fase 1:

- Definición del alcance del subproceso.
- Definición de responsables.
- Definición del plan piloto de prueba.

Fase 2:

- Capacitación previa al inicio del plan piloto.
- Puesta en marcha de plan piloto.
- Cierre del plan piloto de prueba.

Fase 3:

- Capacitación para miembros fuera del piloto.
- Paso a producción de todo el modelo rediseñado

Fase**1:**

Definición del alcance del subproceso: El alcance de este proyecto incluye la aplicación del subproceso de pruebas rediseñado dentro del desarrollo regular de la empresa mediante un plan piloto. La implementación de este se hará a manera de prueba y se controlará de forma manual ya que, del resultado de esta prueba, la empresa tomará la decisión de invertir recursos para la implementación de un sistema que automatice el control del mismo.

Definición de responsables: Se definen como responsables de la implementación del proyecto a:

- Pablo Barba (Monitoreo y control)
- Jeannette Gurumendi (Monitoreo y control)
- George Harris (Product Owner, líder del grupo)

Definición del plan piloto de pruebas:

El piloto de prueba se realiza incluyendo únicamente a un grupo equivalente al 30% del total de miembros involucrados en el subproceso regular, que serían los miembros del equipo de desarrollo, pruebas, cliente y los responsables de la implementación mencionados anteriormente.

El piloto de pruebas tendrá una duración de 2 Sprints consecutivos completos (30 días por cada Sprint, 60 días en total) y será el proceso base que regirá el desarrollo de la empresa durante este tiempo.

Dado que el rediseño del subproceso de pruebas incluye un importante número de cambios, hace muy difícil que se pueda incluir todas las actividades del rediseño en el piloto de prueba, por lo cual, hemos tomado algunas de las actividades más críticas del subproceso rediseñado para que puedan ser efectivamente probadas en este piloto. Las actividades que no han sido incluidas para el piloto de prueba son:

- Considerar todos los puntos de vista (Todo el equipo)
- Generar ejemplos de alto nivel (Equipo cliente)
- Escribir tarjetas de tareas de pruebas (Equipo de pruebas)
- Escribir tarjetas de tareas de desarrollo (Equipo de desarrollo)
- Unificar tarjetas de tareas (Equipo de pruebas)
- Realizar pruebas end-to-end (Equipo de pruebas)

Estas actividades podrán ser incluidas en futuros Sprint de ser requerido por la empresa.

En el Anexo 3 se puede ver un modelo que muestra las actividades que se han incluido en el piloto de prueba.

Fase2:

Introducción y capacitación

En este punto, se debe presentar a los miembros de los diferentes equipos cómo es el subproceso de pruebas rediseñado y cuál será el rol y responsabilidades de cada miembro respecto al mismo. Al presentar el subproceso nuevo se dan a conocer cuáles son las mejoras en comparación con el subproceso actual y cuáles serían los objetivos que esperamos alcanzar con el mismo.

La capacitación implica revisar con más detalle cada una de las actividades que se van a realizar con los involucrados de cada equipo, de esta forma nos aseguramos de que todos tengan claro el desarrollo de las mismas y se pueda seguir con el flujo establecido en el subproceso rediseñado. Cabe indicar que no se realizarán cambios a la propuesta una vez iniciada la implementación.

Puesta en marcha

Para la puesta en marcha se consideran los siguientes puntos:

- La implementación del subproceso rediseñado empieza junto con el inicio de un Sprint regular de la empresa.

- El desarrollo que se haga durante la prueba del nuevo subproceso de pruebas no tendrá ninguna particularidad fuera del trabajo que se realiza regularmente.

Estando los diferentes equipos capacitados respecto a lo que el rediseño del subproceso implica se procede a incluir este en el proceso de desarrollo de la empresa. El monitoreo de cumplimiento del subproceso se hará constantemente y estará a cargo de los responsables del proyecto, incluyendo una reunión de seguimiento con frecuencia diaria en conjunto con todo el equipo de la siguiente forma:

Cierre

Para el cierre del ejercicio de prueba del subproceso de pruebas rediseñado, los responsables deben presentar un informe con los datos que se han podido obtener de los dos Sprints en los que se trabajó. El informe incluye el estado de los objetivos planteados y los valores de las métricas asociadas, así también un resumen de cuáles fueron los puntos altos y bajos de la implementación del proceso rediseñado que son discutidos con todo el equipo.

Fase 3:

Capacitación para miembros fuera del piloto

En este punto, la empresa tomaría la decisión de avanzar con la puesta en marcha de todo el subproceso de pruebas rediseñado. Habiendo realizado un piloto con un grupo de colaboradores, es necesario realizar una capacitación

que incluya al resto de involucrados en el proceso para que puedan avanzar con la implementación sin problemas.

Paso a producción de todo el modelo rediseñado

Una vez que todo el equipo ha recibido la capacitación, se procede a incluir las actividades que fueron excluidas del piloto como parte del proceso regular y a implementar los respectivos controles y mediciones que son necesarios a lo largo del ciclo.

6.2 ANÁLISIS DE RESULTADOS

Siguiendo con el ciclo BPM, en este punto hacemos referencia al monitoreo del proceso. Hemos realizado análisis de los resultados por sprint, esto quiere decir, que terminado cada sprint hemos reunido al equipo y revisado la realización del proceso.

Sprint 1 con el proceso mejorado

Por ser el primer sprint con el proceso mejorado es ya poner en práctica todo lo visto en las capacitaciones y llevar a cabo las actividades indicadas en el nuevo proceso.

De acuerdo con la perspectiva del equipo con respecto a todo el proceso de desarrollo de nuevas funcionalidades con respecto a los principios ágiles llegamos al resultado indicado en la siguiente tabla

Tabla 36 Agilidad por área Sprint 1

Agilidad por área	%
Equipo	33%
Backlog	100%
Daily Scrum	96%
Sprint	72%
Codificación	64%
Pruebas	60%
Negocio	68%
Retrospectivas	60%
Puntuación del Equipo:	128
Total Porcentual:	69%

Lo que se pudo apreciar, se están realizando actividades claves de la agilidad que antes no se realizaban especialmente en el equipo de pruebas donde todos coincidieron que se ve una coordinación del equipo con los equipos de desarrollo y cliente.

En este primer sprint no se tuvo la participación del equipo de clientes en las pruebas de aceptación. Se realizaron algunas pruebas con el Product Owner, pero no se llegó a realizar todas las verificaciones planificadas.

El equipo de pruebas no pudo tener a tiempo los casos de pruebas para la realización de las pruebas unitarias, lo que se decidió realizar la automatización de solo una nueva funcionalidad pequeña para iniciar con el proceso.

El equipo de desarrollo se acopló sin mayores problemas a la implementación del TDD en esa nueva funcionalidad pequeña, pero el problema que se presentó fue que se empleó más tiempo de implementación más de lo estimado.

Con respecto a la cantidad de funcionalidades realizadas dentro del sprint, esta llegó a hacer de 8 puntos para este primer sprint. Se tomó más tiempo de lo estimado por la aplicación de TDD para el desarrollo de pruebas unitarias. Por otro lado, el equipo de pruebas se demoró en la realización de casos de pruebas por demora en obtener ejemplos de las historias de usuarios porque no disponía del tiempo del equipo cliente.

Con respecto a los errores reportados dentro del sprint fueron de acuerdo con la siguiente tabla:

Tabla 37 Errores dentro del sprint 1

Origen de Errores	Errores
Historias no completas	4
Codificación	25
Casos de pruebas no considerados	2
Total	31

La medición de los errores y su origen se comenzó a llevar para poder detectar los puntos de mejoras del proceso y para controlar que el proceso va dando resultado siguiendo la idea de prevenir los defectos en lugar de encontrarlos.

Mientras se realizaba el segundo sprint, se pudo obtener retroalimentación de los clientes, y reportaron 3 errores en producción de este primer sprint. Lo que indica que aún se tienen errores en producción.

También se comenzó a llevar la métrica de automatización de casos de pruebas para ir controlando el incremento y mejora del equipo con respecto a la automatización. En este primer sprint se realizó la automatización de pruebas unitarias de una nueva funcionalidad pequeña.

Durante la retrospectiva se llegaron a definir las siguientes mejoras:

- Realizar lista de chequeo de pruebas comunes en cualquier funcionalidad para que ayude al equipo de desarrollo mientras pruebas termina de realizar los casos de pruebas.
- Verificar con todo el equipo las historias de usuarios completas, porque solo pruebas estaban como responsable de realizarlo.
- Definir fechas anticipadas para coordinar las pruebas de aceptación por parte del equipo de cliente.

Como se puede apreciar, las mejoras no son tanto al subproceso de pruebas sino a como realizar mejor dicha actividad.

Sprint 2 con el proceso mejorado

Luego de la finalización del segundo sprint, nuevamente el equipo se reunió a evaluar el proceso llevado.

Con respecto a la agilidad, el equipo evaluó lo siguiente:

Tabla 38 Agilidad Sprint 2

Agilidad por área	%
Equipo	67%
Backlog	100%
Daily Scrum	96%
Sprint	72%
Codificación	72%
Pruebas	70%
Negocio	72%
Retrospectivas	68%
Puntuación del Equipo:	141
Total Porcentual:	76%

Se ve que con respecto a las áreas de equipo, pruebas y negocio se ha incrementado satisfactoriamente, esto da una visión clara que todo el equipo está percibiendo el cambio de cultura y actitud, trabajando más colaborativamente. Esto permitirá al equipo a llegar a muchos más retos y mejora continua.

En este segundo sprint existió más participación del equipo cliente, aunque no fue todo lo requerido, se pudo ver su participación principalmente en los nuevos requerimientos con mayor riesgo de aceptación por parte de los clientes. Esto ayudó al equipo de pruebas a tener a tiempo los casos de pruebas y ejemplos de las historias de usuarios que necesitaban.

El equipo de desarrollo pudo realizar un poco de implementación de pruebas unitarias automatizadas. Esta mejora se va hacer incremental a medida que el equipo gane experiencia y habilidades en las herramientas de automatización.

Con la realización de listas de chequeos y adicionar los casos de pruebas a nivel general, el equipo de desarrollo pudo avanzar el desarrollo de TDD, aunque no fueron todas las funcionalidades, pero se ve una mejora paulatina del mismo.

En este sprint también se implementó la técnica de pares, esto quiere decir que un desarrollador trabaja con un Tester en la revisión durante la codificación, lo cual fue muy provecho ya que esta funcionalidad no tuvo errores luego de pasado a pruebas ni en las pruebas de aceptación, por lo que el equipo decidió implementar esta técnica para las funcionalidades más complejas y con riesgos.

El equipo de pruebas definió con el equipo de clientes fechas fijas para revisión de pruebas de aceptación y les fue mucho mejor en esa actividad.

Con respecto a errores en producción se reportaron 2 errores en producción de este segundo sprint. Esto permite ver que está bajando y permite analizar que mejoras realizar para seguir con esta tendencia.

Se llegaron a realizar 10 puntos al mes, este valor está muy cerca al realizado en meses anteriores sin la mejora del proceso, lo que nos lleva a pensar que se puede llegar a incrementar la cantidad de puntos realizados en el mes y con mejor calidad.

Los errores encontrados dentro del sprint se presentaron de acuerdo con la siguiente tabla:

Tabla 39 Errores dentro sprint 2

Origen de Errores	Errores
Historias no completas	2
Codificación	14
Casos de pruebas no considerados	2
Total	18

Como vemos el total de errores encontrados dentro del sprint bajó con respecto al sprint 1, la idea de prevenir errores en lugar de encontrarlos mejoró principalmente porque el equipo de desarrollo y pruebas trabajaron juntos en la etapa de codificación probando y codificando mientras se realiza la nueva funcionalidad.

Con respecto a la métrica de automatización podemos indicar que se incrementó en una funcionalidad más con respecto al sprint anterior, lo que nos lleva a pensar que poco a poco el equipo va ganando las habilidades con el uso de las herramientas de automatización.

El equipo llegó a las siguientes mejoras para realizar en el siguiente sprint:

- Iniciar la implementación de las historias de usuarios cuando se encuentre con todos los ejemplos y criterios de aceptación listos.
- Los criterios de aceptación deben ser aprobados por el equipo cliente, una vez que se encuentren realizados. Con esto se evita el incremento de errores de origen historia de usuarios.
- Para evitar incrementar errores en producción, el equipo de pruebas realizará pruebas exploratorias de 2 horas máximo antes de que el equipo cliente realice las pruebas de aceptación finales.

Resultado

Una vez analizado los sprint, podemos llegar a los siguientes resultados de la aplicación del proceso mejorado.

- Se ha incrementado la aplicación de la Agilidad en el equipo en general.
- Hay más trabajo colaborativo entre los equipos. Se vio claramente reflejado en el sprint 2 al aplicar la estrategia de pruebas de pares.
- Aun no se encuentra en un cien por ciento el involucramiento del equipo cliente, pero se pudo apreciar que se encuentra en el proceso de realizarlo.

- El equipo de pruebas tiene clara las actividades que debe realizar en el sprint, aunque aún les cuesta en poco realizarlas todas, pero es un camino que se va a ir completando en los siguientes sprints.
- Las actividades del equipo de pruebas con el equipo de desarrollo han sido realizadas con éxito, aunque en el primer sprint les costó tener los casos de pruebas listos, ya en el siguiente sprint se organizaron mejor.
- Las actividades del equipo de pruebas con el equipo cliente, aún falta por realizarlas mejor porque aún no se cuenta con ellos un cien por ciento involucrados.
- Con respecto a la calidad entregada a los clientes, podemos darnos cuenta que los errores en producción han bajado.

Con respecto al proceso mejorado, el equipo considera que se encuentran correctas las actividades indicadas y lo que se debe ir puliendo son los detalles dentro de las actividades.

6.3 COMPARACIÓN DE MÉTRICAS

Concluido las pruebas del nuevo subproceso de pruebas con los dos sprint, vamos a comparar las mediciones de sprint anteriores y las vamos a comparar con las mediciones obtenidas en los sprint de pruebas.

Para poder comparar las métricas vamos a llevar las métricas de sprints anteriores a la misma proporción que el equipo formado para el plan piloto. Esto quiere decir, al equivalente al equipo piloto.

Medición del grado de madurez de Agilidad del Equipo

En la figura 6.1 se compara la medición de la agilidad del equipo con las evaluaciones luego de cada sprint del plan piloto. El mayor incremento lo ha tenido el área de equipo. El nuevo subproceso de pruebas promueve mucho la colaboración del equipo para obtener resultados de calidad.

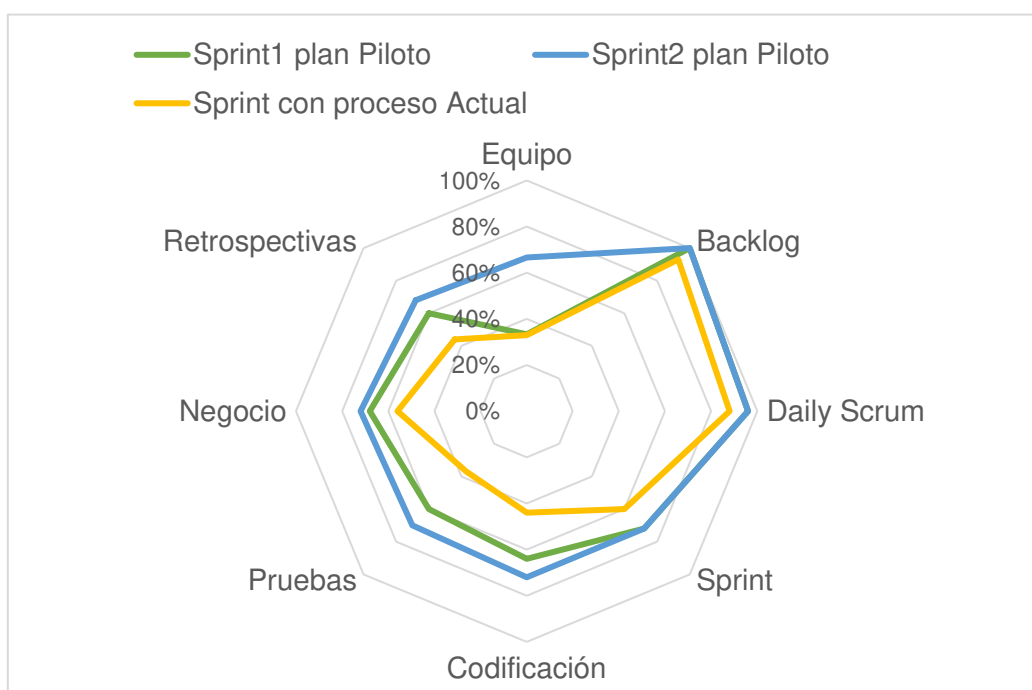


Figura 6.1 Comparación de Evaluación de la agilidad

La segunda área de incremento es pruebas, el equipo de pruebas conoce mejor sus actividades desde el inicio del proceso.

La figura 6.2 compara la evaluación inicial de agilidad con las evaluaciones resultantes al final de cada sprint de prueba, lo que permite evidenciar que ha

existido una mejora del 20% debido a la aplicación de los lineamientos del Scrum Master de la mano con las mejores prácticas del Agile Testing.

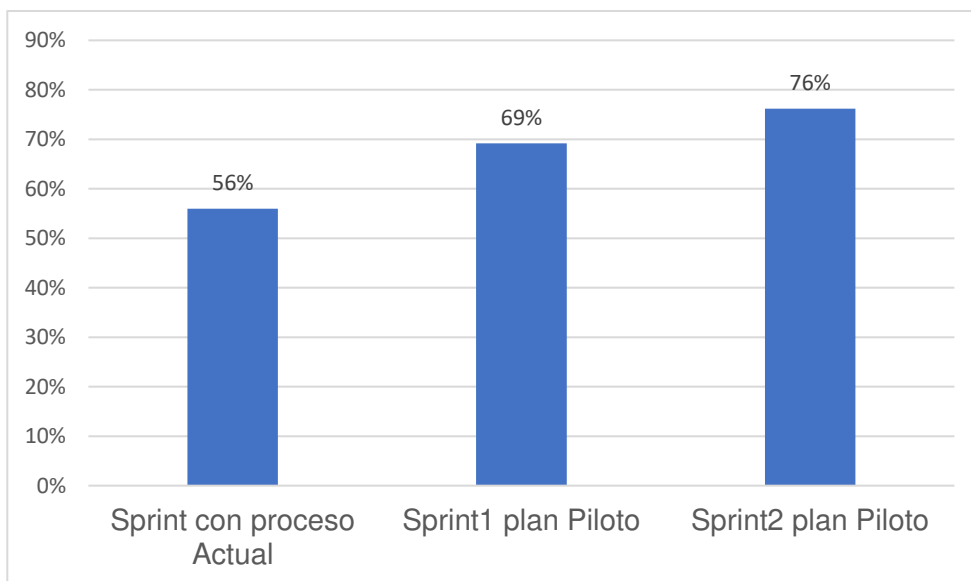


Figura 6.2 Comparación agilidad

Medición de Error en clientes

Hemos venido midiendo la cantidad de errores que encuentran los clientes en un sprint liberado para poder ir tomando acciones correctivas. Actualmente se tiene un promedio de 4 errores mensuales producto de las nuevas funcionalidades entregadas en un equipo como el sprint del plan piloto.

La figura 6.3, se ve que la tendencia de la cantidad de errores en producción, producto del sprint liberado, es a la baja.



Figura 6.3 Comparación errores en producción

La mejora corresponde a una disminución de errores en producción del 50% tomando el resultado del último sprint con el proceso mejorado con relación al sprint con el proceso actual. Podemos concluir que la aplicación de las estrategias de agile Testing dan resultado.

Medición de puntos de liberación por sprint

El equipo actualmente viene realizando un promedio de 16 puntos por Sprint que representa nuevas funcionalidades desde su planificación hasta su liberación en un equipo como el planteado en el plan piloto.

Como vemos en la figura 6.4 se nota una baja del 37,5% de la cantidad de puntos liberados, lo cual lo vimos como normal porque el equipo baja su productividad cuando inicia un proceso nuevo, era uno de los impactos esperados.

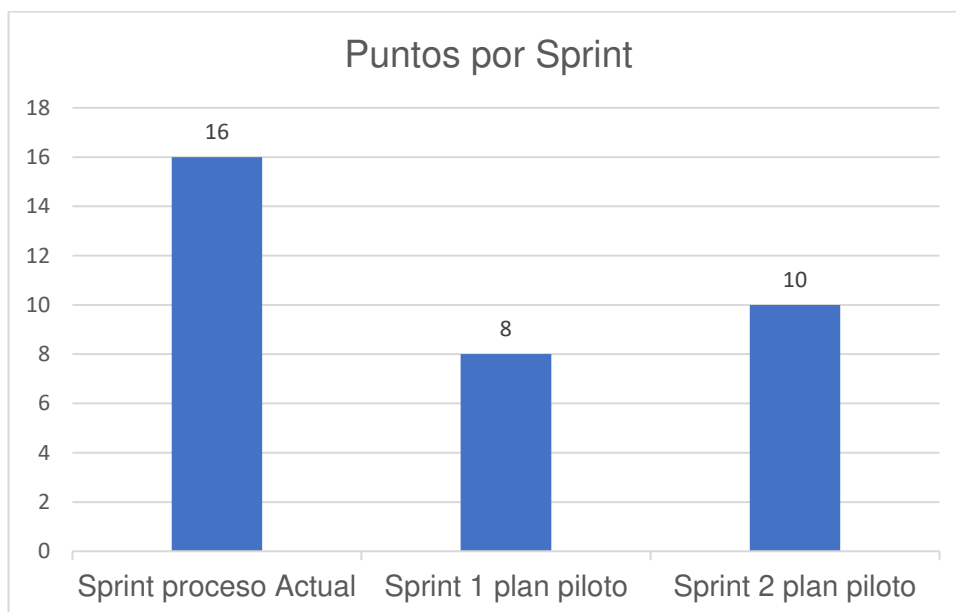


Figura 6.4 Comparación puntos por sprint

También el uso de nuevas herramientas para automatización de pruebas disminuye la velocidad del equipo hasta que llegue a su punto de equilibrio que esperamos que sea en 6 meses.

Las dos métricas anteriores nos permiten considerar que se ha ganado el 50% de mejora de la calidad y se impacta un promedio de 37,5% de la productividad. La productividad se va a ir aumentando un 20% con lo que podemos concluir que se va ganando calidad y la productividad se va ha ir mejorando en el tiempo que a largo plazo genera ganancia para la empresa por el impacto positivo que será tener un producto de buena calidad y satisfacción de los clientes.

Siguiendo con las métricas, hemos propuesto llevar dos métricas nuevas que son: Medición de errores dentro del Sprint y Casos de pruebas automatizados.

Medición de errores dentro del Sprint

De acuerdo las recomendaciones de Agile Testing, el encontrar errores al momento de realizar la ejecución de pruebas es más costoso que encontrarlos antes, por lo que es preferible prevenirlos en actividades anteriores.

Mientras baja la cantidad de errores por sprint, quiere decir que el equipo está trabajando más en prevenirlos.

La figura 6.5 muestra los primeros valores recopilados en los sprints de pruebas, donde se muestra que baja del sprint 1 al sprint 2, y la idea es seguir controlando estos valores y mejorando.

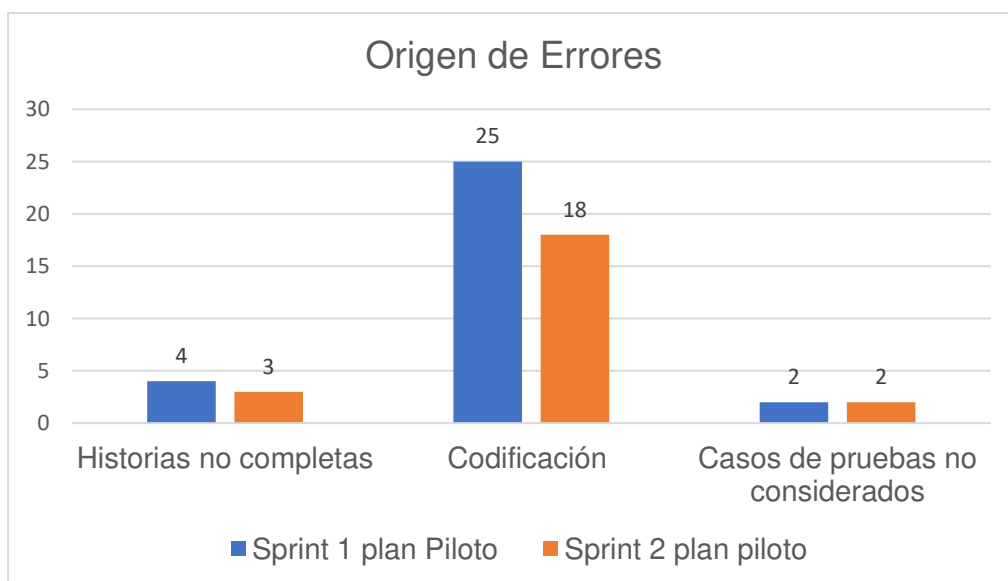


Figura 6.5 Métrica Origen de errores dentro del sprint

Medición de automatización de casos de pruebas

Como hemos visto la automatización de pruebas es una herramienta que permite mejorar en la calidad y la prevención de errores luego de un cambio en la codificación. Esta métrica permitirá ir conociendo el avance de la automatización de pruebas unitarias y en las demás automatizaciones que se realicen en el futuro.

Durante los sprint de pruebas se realizó poco la automatización, pero va a ir incrementando a medida que el equipo vaya ganando habilidades en las herramientas de automatización y por lo cual es muy importante ir midiendo este avance.

En la siguiente figura se muestra el resultado los sprints de la métrica de automatización en el plan piloto.

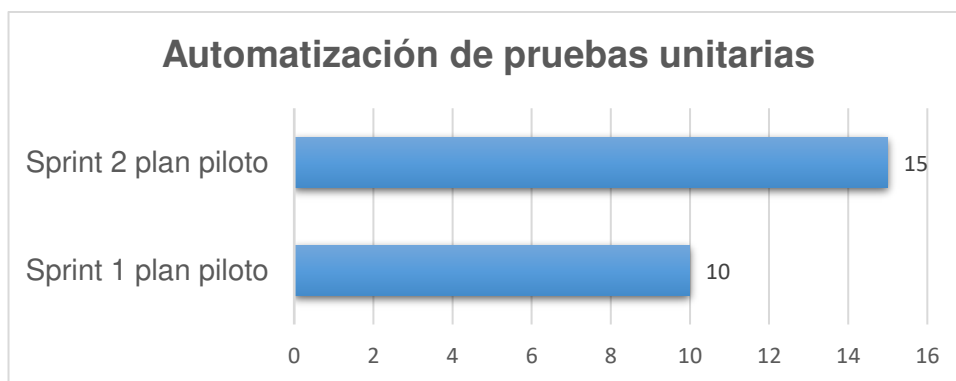


Figura 6.6 Métrica de Automatización de pruebas unitarias

El equipo pudo avanzar poco en la realización de pruebas unitarias, no es fácil a adaptación de la técnica TDD no solo por las herramientas, sino también por el cambio de la manera de programar.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

1. Al realizar el análisis inicial de los problemas de calidad del Software, el equipo pudo darse cuenta de que realmente la cultura ágil no ha sido adoptada completamente dentro del equipo de desarrollo, y mucho menos el equipo de pruebas que no había sido involucrado en esta cultura.
2. El análisis de las mejores prácticas del Agile Testing permitieron definir nuevas actividades para el equipo de pruebas y su trabajo colaborativo con los equipos de desarrollo y cliente.
3. Mediante la aplicación de BPM se realizó la documentación del proceso actual y permitió analizar y mejorar del subproceso de pruebas.

4. El análisis de los sprints de pruebas ayudó a confirmar que mediante las prácticas y actividades recomendadas por el Agile Testing, los equipos incrementarían la colaboración entre ellos y mejoran la calidad del producto.

RECOMENDACIONES

1. Los resultados positivos obtenidos en los sprints de pruebas nos permiten recomendar seguir con el subproceso de pruebas mejorado en los futuros sprints.
2. Seguir evaluando las métricas y mejorar el proceso o las actividades de acuerdo a las mejoras propuestas por el mismo equipo.
3. La cultura del equipo debe seguir incrementando y encaminando al Agilismo, para ello recomendamos un Agile Coach que ayude al equipo a no salirse del camino de la agilidad y calidad.
4. Realizar análisis luego de un año de aplicación del nuevo subproceso de pruebas y a la par de todo el proceso de desarrollo para ir encaminando al equipo al proceso DEVOPS.

BIBLIOGRAFÍA

- [1] TMcLellan, Process Transformation, NA: ABPMP, 2013, pp. 1-8.

- [2] M. Dumas, M. La Rosa, J. Mendling y H. A. Reijers, Fundamentals of business process management: Second Edition, 2018, pp. 1-527.

- [3] A. Martel, Gestión práctica de proyectos con Scrum: desarrollo de software ágil para el Scrum Master, Wroclaw: s.n., 2016.

- [4] L. G. Definitiva, «La Guía de Scrum TM».

- [5] J. Gregory y L. Crispin, Agile Testing Condensed, leanpub, 2019.

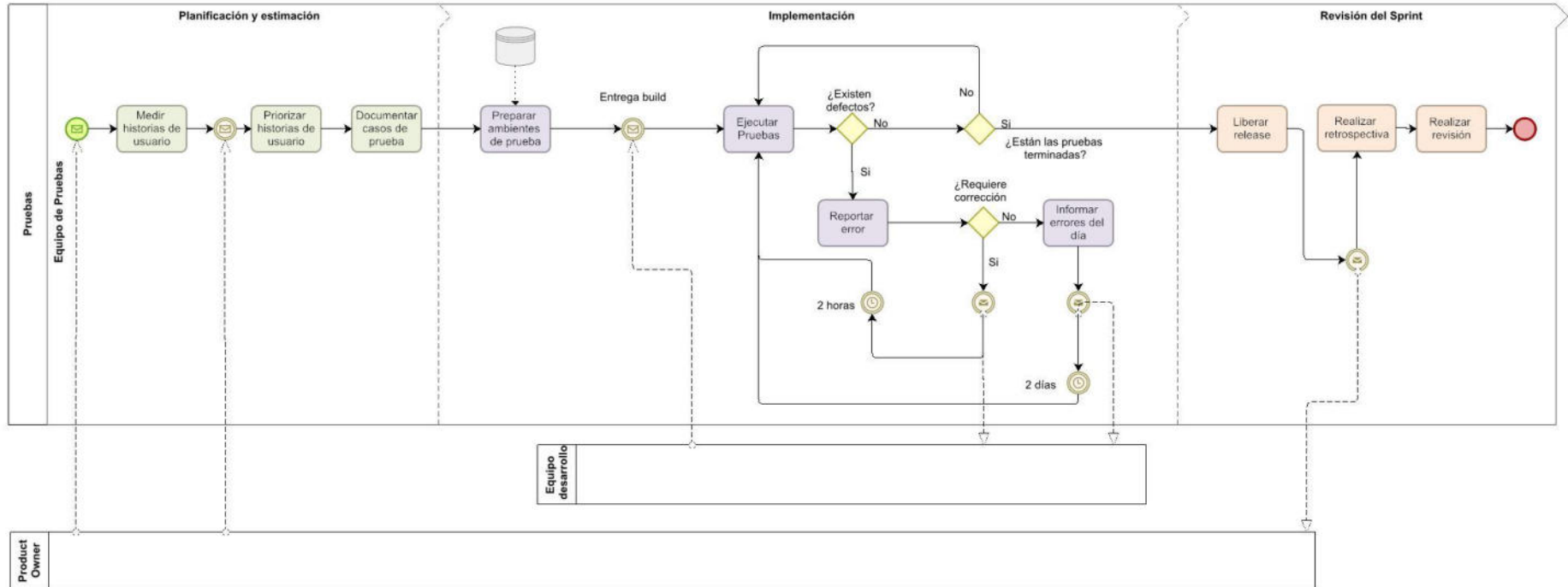
- [6] . Laing Samantha y . Greaves Karen, Growing Agile: A Coach's Guide to Agile Testing, 2015, pp. 1-65.

- [7] L. Crispin y J. Gregory, Agile testing: a practical guide for testers and agile teams, Upper Saddle River, NJ: Addison-Wesley, 2009, p. 533.

- [8] J. Gregory y L. Crispin, More Agile Testing: Learning Journeys for the Whole Team, 2014.

ANEXOS

ANEXO 1: MODELO AS-IS DEL SUBPROCESO DE PRUEBAS



ANEXO 2: MODELO TO BE DEL SUBPROCESO DE PRUEBAS

