PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS

DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SISTEMAS Y COMPUTACIÓN

"DESARROLLO DE UNA GUIA DE UN MARCO DE REFERENCIA DE CALIDAD PARA LA METODOLOGÍA DE DESARROLLO DE SOFTWARE ÁGIL SCRUM"

ANDRÉS RENATO ROJAS PAVÓN

DIRECTOR: ING. FABIÁN DE LA CRUZ

QUITO, AGOSTO 2016

DEDICATORIA

Papá Rodri, mi ejemplo a seguir, mi héroe eterno. Si no fuera por usted, nada de esto fuera posible. Le extraño mucho, que Diosito le bendiga siempre. No me alcanzará la vida entera para agradecerle todo lo que hizo por mí, por ello todo el esfuerzo y dedicación de mi carrera y mi vida, es para usted.

AGRADECIMIENTO

Quiero agradecer a Dios por todas sus bendiciones y por poner en mi vida a todas las maravillosas personas que llenan mi corazón y que de una u otra forma han hecho de mi lo que soy ahora.

Agradecerle a mi Mami que siempre supo estar ahí para mí, siendo un ángel guardián, la mejor madre del mundo, a quien amo y admiro profundamente por todo lo que ella es y representa en mi vida. A mis ñaños, quienes siempre me alentaron a seguir luchando y por ellos busco superarme diariamente, los llevo siempre en mi corazón. A mis abuelitos, a mis tíos y a mis primitos que ni la distancia ha logrado separarnos ni menguar el inmenso amor que existe entre nosotros.

A Javier por su incondicional apoyo e impulso a seguir creciendo profesionalmente. A los ingenieros Fabián De la Cruz, Oswaldo Espinoza y Suyi Arcos que supieron guiarme en los últimos pasos de mi vida universitaria.

A mis amigos Pablo y Pepe que con tantos años de conocernos y todo lo que hemos vivido, ya son mis hermanos por siempre. Y de una manera muy especial quiero agradecer a mi Alejita, quien siempre estuvo conmigo compartiendo toda la carrera, apoyándome e impulsándome a seguir adelante, dentro y fuera de la universidad siempre has sido mi mejor compañía y mi mayor amor.

CONTENIDO

Capítulo. I: INTRODUCCIÓN

En el presente capítulo se abren los temas relacionados con la problemática de la necesidad de un marco de referencia para conducir el proceso de pruebas dentro de la Metodología de desarrollo Ágil Scrum, al igual que sientan los antecedentes y justificaciones para la propuesta de Desarrollo de una Guía de Marco de Referencia para la metodología antes mencionada.

1.1. Introducción

El Instituto de Ingeniería Eléctrica y Electrónica o IEEE (por sus siglas en Ingles) en su estándar IEEE Std 610, define a la calidad como el grado en el cual un componente, sistema o proceso satisface requisitos específicos y/o necesidades y expectativas del usuario/cliente[1].

La calidad del software puede verse reflejada en mayor o menor grado de acuerdo a la importancia del software, tanto en su utilidad y área de trabajo. Es así que como ejemplo contemplemos dos escenarios: el software de facturación en un supermercado y el software de control a bordo del **Ariane 5**[2] en su vuelo 501.

En el primer escenario, si el sistema de facturación de un supermercado presentara un fallo por la caída del sistema o un error al procesar la orden de compra, seguramente se darían inconvenientes al momento de atender a los clientes, pero después de todo, los inconvenientes causados por la caída del sistema no tendrían mayor trascendencia en los clientes, quienes muy probablemente acudan a otro supermercado o regresen más tarde.

En cuanto al segundo escenario, el Ariane 5 fue un cohete de la empresa **Arianespace**, lanzado 4 de junio de 1996, que, a causa de un fallo en el software de control a bordo, a los pocos segundos de su lanzamiento, éste tomo un giro inesperado en su trayectoria, dando la vuelta y explotando. Las consecuencias del fallo del software fueron mucho más grandes, cuantiosas pérdidas económicas, tiempo invertido perdido y pudo haber sido un vuelo tripulado donde hubiera habido pérdida de vidas humanas que lamentar (Ibáñez, 2014).

[1] (Xplore, 2012)
[2] (Arianespace, 2015)

Efectivamente, la gravedad de un fallo de software va a depender enteramente del entorno en que dicho software sea empleado; pero esto no quita que, aunque el ámbito en donde trabaje el software se de bajo impacto, el software deba tener un nivel de desempeño y fiabilidad que garantice al usuario el poder contar en todo momento con su software, ya sea en la atención a un negocio pequeño o en un proyecto multimillonario. Y aún más, si de este software dependen la salud o la vida de personas.

1.2. Situación Actual

En el mundo del desarrollo del software existen múltiples metodologías que guían todo el ciclo de vida del software. Las distintas metodologías de desarrollo de software tienen diversos enfoques que les permiten ser adoptadas por un sin número de tipos distintos de organizaciones. Hay metodologías rígidas y monolíticas como TSP o RUP llamadas las **metodologías tradicionales** y por el otro lado hay metodologías mucho más flexibles y adaptables a cambios llamadas **metodologías ágiles** tales como Scrum o Kanban.

Dentro de las metodologías llamadas tradicionales, podemos tomar como ejemplo a TSP 'Team Software Process' (Carnegie Mellon, 2010). Una metodología monolítica llamada tradicional con su enfoque en el desarrollo del software en equipos, donde cada miembro del equipo cumple un rol y actividades específicas que responden a responsabilidades propias de cada rol.

Esta metodología de desarrollo, TSP, plantea un esquema para el proceso de pruebas y control de calidad, pero como es característico de una metodología formal, este proceso de pruebas maneja un enfoque muy detallado, acarreando mucha documentación y artefactos tales como Planes de Pruebas, Procedimientos para la realización de las mismas, entre otros; y de esta manera se lleva un control minucioso de lo que se planea realizar y de lo realizado dentro del proceso de pruebas para el control de la calidad del software. Todos estos artefactos junto con Diagramas y Modelos de datos son documentos que sirven como evidencia y respaldo de lo realizado durante el desarrollo del software.

Por otro lado, las metodologías ágiles, como es el caso de la metodología Scrum, desarrollada en los años 90's por Jeff Sutherland y Ken Schwaber[3], no manejan una documentación suficiente, que conduzca el proceso a través del ciclo de vida del desarrollo del software al igual que no disponen de artefactos tales como Diagramas y/o Casos de Uso, que evidencien el progreso en el proceso de desarrollo.

Los procesos dentro de las metodologías ágiles, como Scrum, se manejan bajo una guía de recomendaciones y buenas prácticas, las cuales son características fundamentales de una metodología adaptable y dinámica, esta guía de las metodologías ágiles, es suficiente para desarrollar y adaptar la misma a cualquier ámbito o equipo de trabajo no necesariamente un equipo de desarrollo de software.

El no hacer necesario contar con documentación y llevar registro explícito de las actividades realizadas y por realizar, es una parte fundamental para manejar una metodología dinámica y adaptativa como lo es Scrum. Esto permite tener la flexibilidad de implementación en el ambiente que sea, y ofrecer frutos inmediatos resultantes del desarrollo de los procesos de la metodología; pero al mismo tiempo esto deja en evidencia una falencia de la metodología ágil en el área de pruebas y control de calidad.

Dicha falencia se enfoca principalmente en la no realización temprana de los planes de pruebas necesarios para el control del desarrollo del software, en no disponer una estandarización de qué pruebas son las recomendadas y necesarias de acuerdo a la exigencia del software a evaluar y no se contempla el orden en que las distintas pruebas se deben aplicar y el punto en el desarrollo del software que las mismas deben ser ejecutadas.

En los últimos años, las metodologías ágiles han tomado un papel importante en el campo de desarrollo de software, debido a sus distintas características como su adaptabilidad, flexibilidad, y como lo dice su nombre agilidad, Por ello, estas metodologías ágiles, van creciendo en demanda de acuerdo al enfoque que los proyectos de desarrollo vayan necesitando para guiar su ciclo de vida.

Por lo antes mencionado, el eje central del presente proyecto de disertación fue el Generar un Marco de Referencia de Calidad para apoyar a la Metodología Ágil de desarrollo de

[3] (Scrumguides.org)

software Scrum. La metodología Scrum fue pensada con el fin de proveer un marco de referencia conciso y al mismo tiempo flexible para el desarrollo incremental y rápido de software.

1.3. Justificación

Siendo un proceso, una sucesión de tareas, que tienen como inicio unas entradas y como fin unas salidas, cuyo objetivo es agregar valor en cada etapa del mismo, es importante contar con un proceso de control de calidad para asegurar el mejor producto final. La calidad es la carta de presentación de cualquier producto, por lo cual contar con un proceso que asegure que este producto final cumple con estándares y normas de calidad se vuelve un objetivo indispensable en el desarrollo del mismo.

Para guiar un proceso es necesario tener un manual, una guía que dé las pautas para seguir un adecuado control que asegure la calidad del producto final, con ello el presente proyecto de disertación propone una Guía de Marco de Referencia de Calidad para la metodología de desarrollo de software ágil Scrum, para apoyar y guiar el proceso de Pruebas dentro del ciclo de vida de desarrollo de un producto de software.

Se hace evidente su necesidad en el punto de que la metodología ágil de desarrollo Scrum, no plantea un esquema o pautas de qué pruebas son las más óptimas o en que instancia del desarrollo del software es necesario aplicar dichas pruebas. Y de igual manera no plantea si se debe planificar o en qué momento se deben realizar los planes de pruebas necesarios, para evaluar el cumplimiento de los objetivos del software a realizar.

Con la ayuda de la Guía de Marco de Referencia de Calidad se plantean lineamientos y guías y un aplicable proceso de evaluación, para que de este modo tener una definición de un flujo a seguir y los lineamientos adecuados que permitan adoptar un proceso de pruebas para obtener un producto de calidad.

1.4. Definición del Tema

En el presente proyecto de disertación, estudiando el proceso de pruebas dentro de la metodología de desarrollo ágil Scrum, se recopiló información y se realizó un análisis del

ciclo de vida del software bajo Scrum, que, a pesar de tener ciertas pautas, normas y buenas prácticas dadas por la metodología, evidencia la necesidad de definir un Marco de Referencia formal para administrar el proceso de calidad del software. Con ese antecedente se planteó una *Propuesta De Marco De Referencia De Calidad*, estructurada y definida, que apoye a la metodología antes mencionada.

Esta propuesta tiene un soporte teórico apoyado en CMMi[4] y en estándares internacionales de calidad, para que de esta manera se impulse a la metodología Scrum, y que la misma brinde una guía adecuada para un proceso de calidad que sea escalable y que permita aportar un mayor valor agregado al producto de software

1.5. Objetivos

1.5.1. *Objetivo General*

Elaborar una Guía de Marco de Referencia de Calidad que apoye al proceso de pruebas dentro del ciclo de vida del desarrollo del software guiado por la metodología de desarrollo ágil Scrum, y de esta manera ayude al mejoramiento continuo de todo el proceso en sí.

1.5.2. *Objetivos Específicos*

- Definir el Marco de Referencia de Calidad que guíe el proceso de pruebas de la metodología ágil Scrum.
- Establecer lineamientos y guías que permitan controlar el proceso de calidad a lo largo del ciclo de vida del desarrollo del software bajo la metodología ágil Scrum.
- Identificar los parámetros de evaluación que permitan determinar el estado, crecimiento y mejora continua del proceso de calidad de desarrollo del software bajo la metodología ágil Scrum.

[4] (Carnegie Mellon, 2010)

Capítulo. II: FUNDAMENTO TEÓRICO

En el presente capitulo se desarrollará todo el fundamento teórico y tecnológico necesario para la presente propuesta de Marco de Referencia de Calidad. Entre la información empleada se describe como la metodología de desarrollo ágil Scrum, administra y guía el proceso de desarrollo bajo sus lineamientos, de igual manera se pone en evidencia como por otro lado la metodología RUP, llamada tradicional, hace necesario el llevar un proceso de desarrollo más riguroso y estricto.

Y para complementar se explican ciertos estándares tanto de ISO como de IEEE que estableces criterios para la calidad y además modelos como CMMi y TMMi que plantean algunos lineamientos que representan un aporte a un proceso de pruebas y calidad.

2.1. Proceso de Desarrollo basado en la metodología ágil Scrum.

Según el portal *ProyectosAgiles.org*, determinan a la *Metodología Ágil Scrum* como: un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para **Trabajar Colaborativamente, en Equipo**, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos[5].

Scrum[6] se caracteriza por manejar entregas parciales periódicas del producto final, manejando constantemente prioridades de estas entregas de acuerdo a las necesidades y exigencias del cliente final del producto. Scrum está recomendado para entornos cambiantes y complejos, donde las necesidades y requerimientos son variables de acuerdo a las prioridades para el negocio.

[5] (Aguiles.org, n.d.)
[6] Glosario de Términos, Scrum

Esta metodología tiene su guía llamada *Scrum Reference Guide*[7], que permite su implementación en una empresa con un enfoque de producción y mucho más aún con un enfoque en el desarrollo de software.

Es así que guiándose con Scrum una empresa de desarrollo de software, bajo esta metodología, puede definir los proceso a seguir para el desarrollo, desde la planificación de las nuevas tareas, desarrollo de las mismas y pruebas[8]; hasta el despliegue en producción (*Deploy*), mensualmente o por períodos de las mismas.

Bajo los parámetros para el desarrollo de *Scrum*, el proceso de la metodología se desarrolla en ciclos de desarrollo fijos llamado *Sprint*, los cuales bajo las buenas prácticas de la metodología se fijan en períodos de entre 7 días mínimo a 30 días máximo, donde el objetivo de cada *Sprint* es presentar un entregable parcial del producto al cliente, susceptible de ser implementado inmediatamente.

El proceso del ciclo comienza con la realización la planificación y priorización de los objetivos requeridos para la presente iteración, tomando en cuenta la importancia y costo que dichos objetivos tengan para el negocio y para el cliente. Esto sucede el primer día de la iteración en la reunión llamada *Sprint Planning*[9], donde basándose en el *Product Backlog*[10], el cual es una lista priorizada, que contiene los objetivos a cumplir en el *Sprint* manifestados por el *Product Owner*[11], se analizan los requerimientos y se fijan las prioridades a tomar en la iteración por comenzar.

Otro paso dentro del proceso de desarrollo *Scrum*, es planificar y realizar una reunión diaria mientras dure el *Sprint*, esta primera reunión llamada *Stand Up Meeting*[12], consiste en encuentros ágiles y cortos, en los cuales cada miembro del equipo da informe al grupo del estado de las tareas que se encuentra realizando. Estas reuniones se las planifica para cada día a la misma hora diariamente y se deben cubrir tres aspectos para el informe al grupo: qué se ha realizado desde el último *Stand Up*, qué se tiene planificado realizar, y qué

[7] (James, 2010-2012)
[8] Glosario de Términos, Pruebas
[9] Glosario de Términos, Sprint Planning
[10] Glosario de Términos, PBI
[11] Glosario de Términos, Product Owner
[12] Glosario de Términos, Stand Up

impedimentos ha encontrado o considera va a encontrar. De esta manera dar un informe de las posibles tareas relacionadas o dependientes y así agilizar el flujo de trabajo del equipo.

Una vez realizada la planificación para el *Sprint* en curso, los objetivos generales para la iteración se sub-dividen en tareas más focales y se las asigna a un miembro del *Equipo Scrum*[13]. Quien será el encargado de desarrollar la o las mismas. Una vez que se culminen las tareas de desarrollo estas son reasignadas a un *Tester*[14], quien en base a la información trasmitida por el desarrollador y/o el *Scrum Master*[15], aplicará las *Pruebas*[16] necesarias a la o las tareas bajo su mejor criterio, experiencia y conocimientos, para asegurar el cumplimiento de las especificaciones y requerimientos iniciales dados por el *Product Owner*.

Posterior a las pruebas realizadas por el *Tester* y si el producto cumple y satisface los requerimientos iniciales, éstas pasan a un proceso de validación y aprobación por parte de equipo y en especial del *Product Owner* en una segunda reunión llamada *Show and Tell*[17], donde se muestra y determina si el entregable parcial del producto ha cumplido con todos los objetivos inicialmente establecidos y el mismo puede pasar a producción, y ser evaluado por el usuario final, o si necesita algún trabajo extra para el cumplimiento de los requerimientos.

En la *Ilustración 1*, se presenta gráficamente, lo antes descrito, sobre cómo se maneja el proceso de ciclo de vida de desarrollo de que maneja la metodología ágil Scrum.

[13] Glosario de Términos, Equipo Scrum
[14] Glosario de Términos, Ingeniero en Calidad
[15] Glosario de Términos, Scrum Master
[16] Glosario de Términos, Pruebas
[17] Glosario de Términos, Demostración

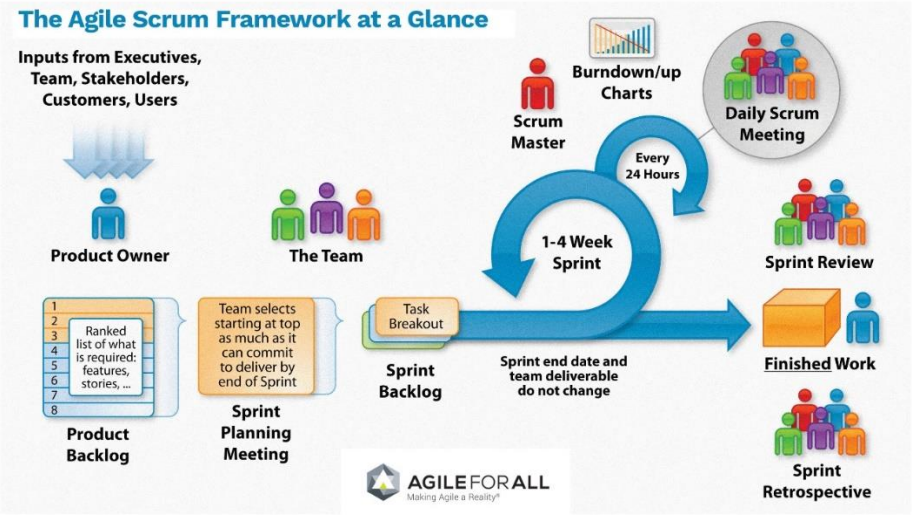

Ilustración 1: ***Diagrama del proceso del Ciclo de vida de Scrum***

Fuente: *(Agile For All, 2016)*

Como *Scrum* es una metodología que maneja un ciclo de vida de desarrollo del software recursivo, el proceso de *Scrum* se vuelve a generar cada inicio de un sprint, en cuanto al desarrollo. Para cumplir con los requerimientos faltantes en anteriores Sprint o para desarrollar una nueva funcionalidad del producto, y de igual manera para corregir la retroalimentación dada en el *Show and Tell.*

Posterior al cierre del *Sprint* el Equipo Scrum mantiene una tercera reunión llamada *Retrospective*[18], donde cada miembro del equipo expresa su criterio frente a lo realizado en el Script enfatizando en lo bueno que se ha conseguido en el trabajo del equipo, y los puntos a mejorar que consideren que ayudarían al crecimiento y mejora del grupo para futuros trabajos.

Así se desarrolla el ciclo de desarrollo de vida del software bajo las buenas prácticas planteadas por la *Metodología ágil Scrum*. Es un ciclo iterativo, que requiere una buena organización, compromiso y trabajo cooperativo de cada miembro para en mutuo apoyo sacar adelante los objetivos aceptados para cumplir con los requerimientos del proyecto.

[18] Glosario de Términos, Retrospectiva

2.2. Proceso de Calidad.

2.2.1. *Metodología Ágil – ¿Qué nos dice Scrum?*

La metodología de desarrollo ágil Scrum, como su nombre lo dice, esta principalmente orientada al proceso de desarrollo del software, sin especificar un proceso de *pruebas* dentro del ciclo de vida de desarrollo del producto de software.

Dentro de las buenas prácticas planteadas por la metodología *Scrum*, no se detalla un proceso de pruebas requerido durante el desarrollo. Por otro lado, los roles dentro de la metodología, de igual manera, no definen un rol de *probador* como tal para el grupo, tan solo engloban en el 'Equipo Scrum' a los miembros del equipo.

Por otro lado, bajo los parámetros de una metodología ágil, *Scrum* no determina qué documentación es necesaria manejar a lo largo del ciclo de desarrollo y mucho menos detalla un plan de pruebas o un proceso de pruebas que se haya seguido o se deba seguir. La metodología pone énfasis en el crecimiento y mejora del grupo de trabajo, pero no analiza la aplicación y mejora de las técnicas aplicadas por el equipo para el desempeño de sus tareas, es así que el proceso de *pruebas* se lo realiza a lo mejor del conocimiento y experiencia individual y no bajo los lineamientos planteados por un proceso de *pruebas* definido.

2.2.2. *Metodología Tradicional – ¿Qué esperar de RUP?*

RUP o Rational Unified Process, por sus siglas en inglés, es un modelo de desarrollo de software enfocado principalmente en ofrecer mejores prácticas de software para desarrollar productos de calidad. (Rational Software- The Software Development Company, 1998).

RUP al ser un modelo tradicional, plantea un proceso de desarrollo definido basándose en seis buenas prácticas para guiar dicho proceso, por supuesto sustentando en material de apoyo, herramientas y asesoría técnica que respalden cada fase del proceso de desarrollo.

Como su nombre lo dice, RUP es un modelo racional que utiliza un lenguaje racional, como es UML (Unified Model Process), para definir el proceso que se lleva a cabo a lo largo del

ciclo de vida de desarrollo del software. A continuación, citaré las seis buenas prácticas planteadas por RUP para habilitar eficientemente un desarrollo de alta calidad[19]:

i. Desarrollo itinerante para mitigar riesgos tempranamente en el proyecto
ii. Manejo eficiente de los requerimientos
iii. Modelado Visual para manejar la complejidad del proyecto
iv. Utilizar arquitectura basada en componentes.
v. Verificar la calidad a lo largo de todo el ciclo de vida.
vi. Control de cambios.

RUP plantea mejorar la productividad del desarrollo del software, apoyando el proceso en lineamientos guías para cada etapa del ciclo de vida, empleando plantillas de documentación para cada paso y utilizando herramientas que agiliten el manejar dicha documentación generada.

En la *Ilustración 2*, podemos identificar gráficamente los dos ejes que el proceso de RUP, en su documento guía[20], describe: un eje Dinámico, el Tiempo, sobre el cual se manejan los ciclos, iteraciones y fases que el modelo RUP emplea durante el ciclo de vida de desarrollo del software y, por otro lado; un eje Estático, el cual consta de las actividades, trabajos y artefactos que el modelo propone como complementos a cada etapa del desarrollo del software para que este cumpla un óptimo desempeño.

[19] (Rational Software- The Software Development Company, 1998, pp. 1-2)
[20] (Ibídem, 3)

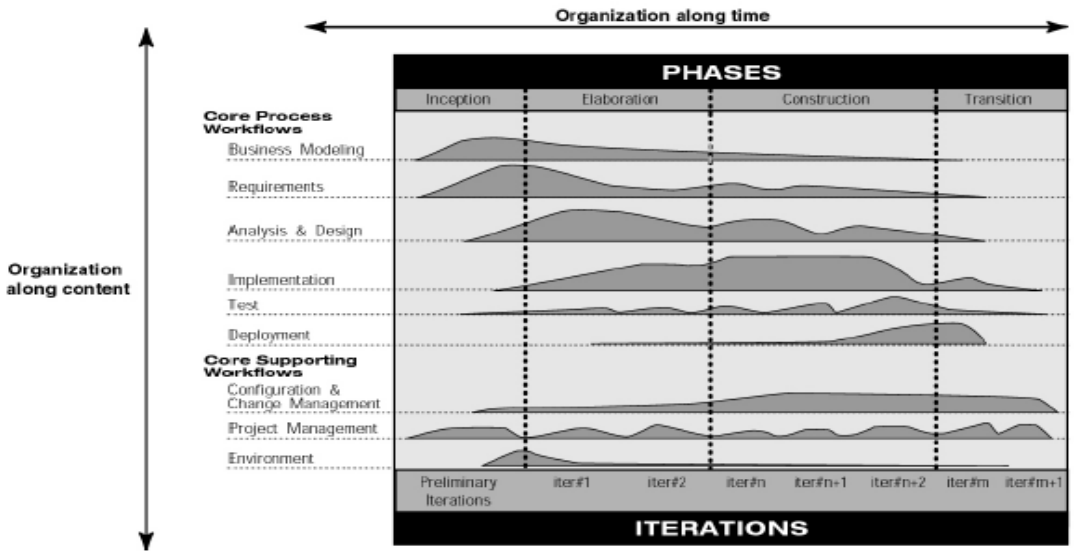

Ilustración 2: ***Modelo iterativo de RUP, muestra como el proceso está estructurado en dos dimensiones.***

Fuente: ***(Rational Software- The Software Development Company, 1998, p. 3)***

EJE DINÁMICO: Organización dinámica del proceso a lo largo del tiempo[21], el eje dinámico de RUP se divide en ciclos, donde en cada uno de los cuales se trabaja y busca generar un entregable del producto, cada ciclo de desarrollo a su vez se divide en cuatro fases consecutivas; cada una de estas fases debe ser bien definida y delimitada, con metas claras que se deban cumplir en un tiempo fijado previamente.

Las fases en el eje dinámico de RUP constan de propósitos específicos que se deben lograr para su cumplimiento, dichas fases son:

- ✓ **Comienzo / Definición:** Durante esta fase inicial, en leve detalle podemos decir que, se define una visión general de los requerimientos del proyecto al igual que los requerimientos claves del mismo, se detalla un modelo inicial de Casos de Uso, se delimita el alcance del proyecto y se establece un plan del proyecto donde se muestren como se van a manejar las fases y las iteraciones del mismo.

  Una vez concluida esta fase, los encargados del proyecto hacen un análisis de costo/beneficio del proyecto, la viabilidad del mismo y toman la decisión si el proyecto continúa o es aplazado y en el peor de los casos se lo cancela.

[21] (Rational Software- The Software Development Company, 1998, p. 3)

- ✓ **Elaboración:** El propósito de esta fase principalmente es analizar la problemática en sí del proyecto, establecer el plan a seguir para el mismo y mitigar los riesgos más altos que se puedan presentar durante el proyecto. Para cumplir esto se debe tener un entendimiento total del sistema, entender sus requerimientos tanto funcionales y no funcionales y tener claro el alcance del mismo.

  Esta fase, se la puede considerar la más fundamental, ya que, para darla por cumplida, toda la parte ingenieril del proyecto debe ser dada por completada, es decir, se debió haber asegurado que los requerimientos, los detalles de la arquitectura y sobre todo la mitigación de los riesgos son lo suficientemente estables como para aprobar el paso a las siguientes fases del ciclo.

- ✓ **Construcción:** Durante esta fase, todos los componentes y características de la aplicación son desarrollados e integrados al producto en desarrollo y todos estos, tanto las características funcionales y no funcionales son probadas profundamente.

  Esta fase es la considerada como el proceso de manufacturación del producto de software donde se enfatiza en el manejo de recursos para a su vez tener una optimización de costos, el tiempo y especialmente de la calidad. Es durante esta fase donde se hace la transición del desarrollo intelectual de las fases de **Definición** y **Elaboración** antes mencionadas, hacia una fase de **Construcción** donde se desarrollará un producto totalmente factible de entregar en las manos de un usuario final.

  Una arquitectura robusta y un plan entendible de desarrollo están altamente relacionados con toda la planificación y definición realizada al comenzar el proyecto. Es decir, una de las cualidades críticas de una arquitectura robusta es una fácil construcción del producto de software. Y así permite determinar si el producto cumple con las características necesarias para ser liberado como una versión “Beta” del mismo.

Es por ello que es crucial el balance entre el desarrollo de la arquitectura y la planeación durante las fases iniciales del proyecto, ya que esta equidad permite desde un inicio estimar y definir hasta el proceso de pruebas que se deben realizar, ya que desde un inicio se conoce las características y funcionalidades que deben de ser probadas para garantizar un nivel de calidad de software.

- ✓ **Transición:** Para esta fase, el objetivo que se persigue es entregar un producto de software lo suficiente maduro, como para que el mismo pueda ser usado por el usuario final. Una vez que el usuario final trabaje sobre el producto entregado aparecerán problemas sobre los cuales se tendrán que tomar medidas, tales como recodificar alguna funcionalidad, completar un requerimiento o modificar alguna funcionalidad, alguna especificación que haya faltado en el inicio.

  Para realizar la transición de entregar una versión del producto de software al usuario final, esta debe cumplir una línea base de madurez que cumpla con determinadas funcionalidades importantes del producto y sobre todo cumpla con un nivel de calidad definido por el proceso de pruebas aplicado durante la fase de **Construcción**. Todo esto acompañado de la documentación necesaria que dé respaldo de todo el proceso realizado.

  La transición del producto al usuario final consistirá en pruebas "Beta", que validarán la versión del producto frente a las expectativas del usuario, apoyándose en los requerimientos iniciales previamente definidos, interoperabilidad entre sistemas, procesamiento de la data por el servidor de base de datos, entrenamientos a los usuarios finales y al personal encargado de dar mantenimiento de primera mano al sistema y liberación del mismo para un proceso de distribución, marketing y comercialización.

  El enfoque de la fase de Transición es realizar todas las actividades necesarias para poner en manos del usuario final un producto de software que cumpla de la mejor manera con todas especificaciones. Esto puede comprenderse en varias iteraciones donde se realicen todas las pruebas necesarias en el ambiente del cliente y efectuadas

por el cliente, al igual que se realice la corrección de problemas encontrados durante las mismas.

Es en este punto de las pruebas se hará necesaria la documentación e informes de los desarrolladores al igual que la toda la documentación desarrollada a lo largo de todas las fases anteriores, para poder brindar un entrenamiento adecuado a los usuarios en cómo funciona y se utiliza el producto de software y de igual manera aportarles los manuales necesarios.

Al final de la fase de Transición, se realiza la evaluación de si se alcanzó los objetivos planteados en el inicio del proyecto, y de si es necesario el comenzar o no un nuevo ciclo de desarrollo para completar los objetivos faltantes o desarrollar nuevos requerimientos.

- ✓ **Iteraciones:** Cada fase del RUP (Rational Unified Process) se lo puede llegar a dividir en iteraciones. Las cuales a su vez consisten en un ciclo completo de desarrollo que involucren las cuatro fases antes descritas (Definición, Elaboración, Construcción y Transición). El resultado de cada iteración será la liberación interna o externa de una versión ejecutable del producto, la cual vaya incrementando de iteración en iteración hasta convertirse en la versión final del producto.

  El uso de iteraciones para el desarrollo de software aporta varios beneficios tales como:

  - ❖ Mitigación temprana de riesgos.
  - ❖ Mejor Manejo de cambios.
  - ❖ Alto nivel de reusabilidad.
  - ❖ Aprendizaje a lo largo de todo el proyecto.
  - ❖ Mejora en la calidad general del producto.

EJE ESTÁTICO: El eje estático de RUP describe **Quién** está haciendo **Qué**, **Cómo** y **Cuándo**. El Proceso Racional Unificado representa esto usando cuatro principales elementos de modelado[22].

1. Trabajadores: "**Quién**"
2. Actividades: "**Qué**"
3. Artefactos: "**Cómo**"
4. Flujo de Trabajo: "**Cuándo**"

- **Trabajadores, Actividades y Artefactos**

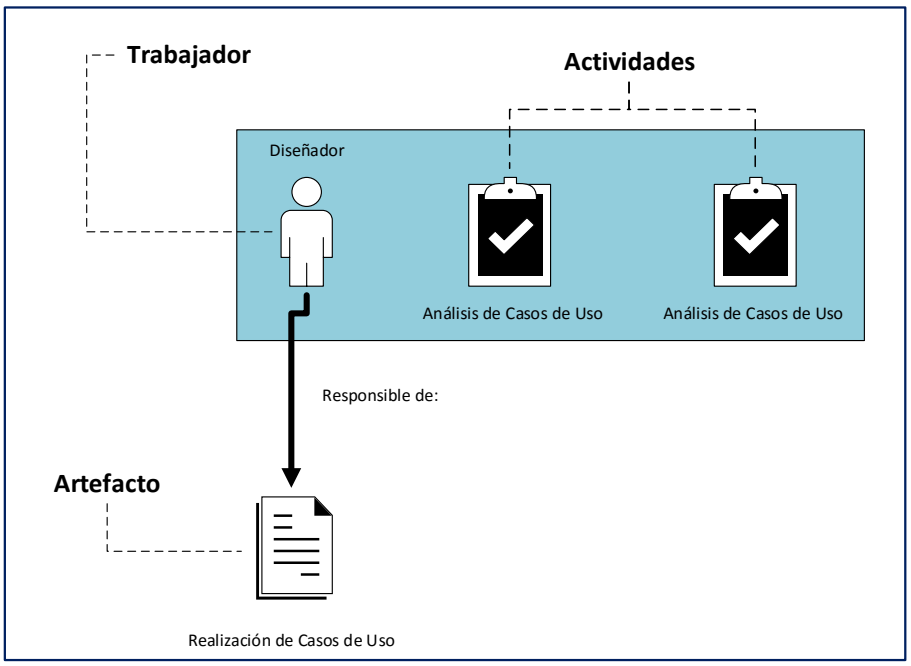

Ilustración 3: ***Relaciones que maneja RUP entre Trabajadores, Actividades y Artefactos***

Fuente: ***(Rational Software- The Software Development Company, 1998, p. 8)***

La *Ilustración 3*, hace referencia a la relación que mantiene el Trabajador frente a las Actividades que el mismo este a cargo y los artefactos que deban ser producidos o actualizados de acuerdo a las actividades correspondientes.

[22] (Rational Software- The Software Development Company, 1998, p. 7)

Trabajadores:

Un trabajo representa el proceder y las responsabilidades que un individuo o grupo de individuos realizan individualmente o como grupo de trabajo en favor del proyecto. Dentro del Proceso Unificado un trabajador, se lo define más como el o los roles que un individuo puede desempeñar dentro del proyecto y que es él quien es responsable de realizar y completar determinadas actividades de acuerdo al rol que desempeñe al igual que es responsable de un conjunto de artefactos que se deben generar.

A continuación, tenemos la *Ilustración 4*, la cual representa algunos trabajos o roles que un individuo puede ser responsable dentro del proyecto al igual que las actividades que debe cumplir bajo el rol que desempeña.

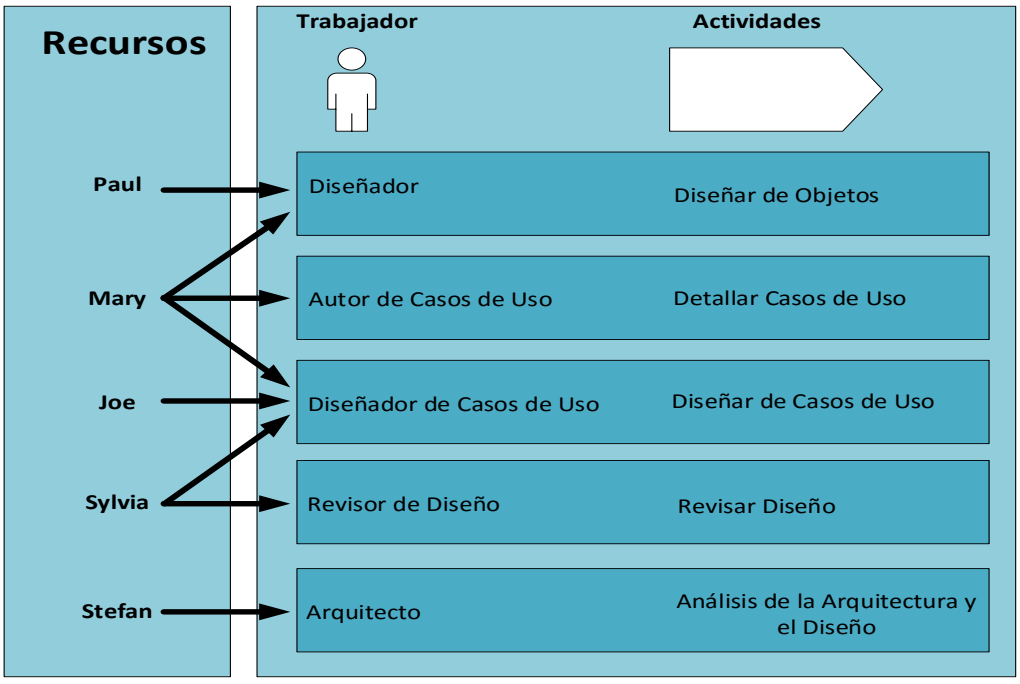

Ilustración 4: ***Relaciones entre Trabajadores y las Actividades manejadas por la metodología RUP***

Fuente: ***(Rational Software- The Software Development Company, 1998, p. 8)***

- **Actividades[23]:**

Una actividad es una unidad de trabajo, que un individuo en el rol que desempeña debe cumplir. Cada actividad tiene un propósito definido como, por ejemplo, crear un artefacto, actualizar un grupo de artefactos como modelos, clases, planes. Para cada actividad es asignado un trabajador específico quien debe cumplir la actividad encomendada en un plazo de horas hasta unos cuantos días, de acuerdo a la complejidad de las tareas.

Esto sirve como unidad de planificación y detalle de progreso para las labores del individuo y del proyecto en sí, donde de acuerdo a la complejidad de las tareas, estas se las planifica individualmente o como parte de una serie de tareas que se deben completar en un período planificado.

Ejemplos de Actividades:

- ✓ **Planificación de Iteración,** Por el Trabajador: *Manager del Proyecto*.
- ✓ **Definir Casos de Uso y Actores,** Por el Trabajador: *Analista del Sistema.*
- ✓ **Revisión de Diseño,** Por el Trabajador: *Revisor de Diseño.*
- ✓ **Ejecución de Pruebas de Rendimiento,** Por el Trabajador: *Probador de Rendimiento*

- **Artefactos:**

Los Artefactos son las piezas de información producidas, modificadas o usadas por el proceso de desarrollo. Los Artefactos vendrían a ser los productos tangibles resultados del proceso, es decir los productos que el proceso genera o usa en el camino a completar el proyecto.

Los artefactos son usados, por los trabajadores, como parámetros de entrada para desarrollar una actividad determinada y a su vez el resultado de esa actividad terminará generando otros entregables o actualizando los artefactos de entrada.

Los Artefactos pueden ser muy variados y definirse de varias maneras:

- ✓ Un modelo: *Modelo de Casos de Uso, Modelo de Diseño.*

[23] (Rational Software- The Software Development Company, 1998, p. 8)

- ✓ Elemento de un Modelo: *Una clase, Un Caso de Uso.*
- ✓ Un Documento: *Documento de Arquitectura de Software.*
- ✓ Código Fuente.
- ✓ Ejecutables.

➢ FLUJOS DE TRABAJO

Conocer los responsables y las actividades a realizar no es suficiente para definir un proceso, como lo vemos en la *Ilustración 5*, es necesario manejar un camino que describa la secuencia que las actividades deben seguir para aportar valor al resultado. UML representa un flujo de trabaja en artefactos tales como diagramas de secuencia y diagramas de colaboración, donde se ponen en evidencia las interacciones entre los trabajadores y la dependencia que tiene cada actividad en el proceso. Esto incluyendo al proceso de pruebas que lo lleva paralelamente al desarrollo en sí.

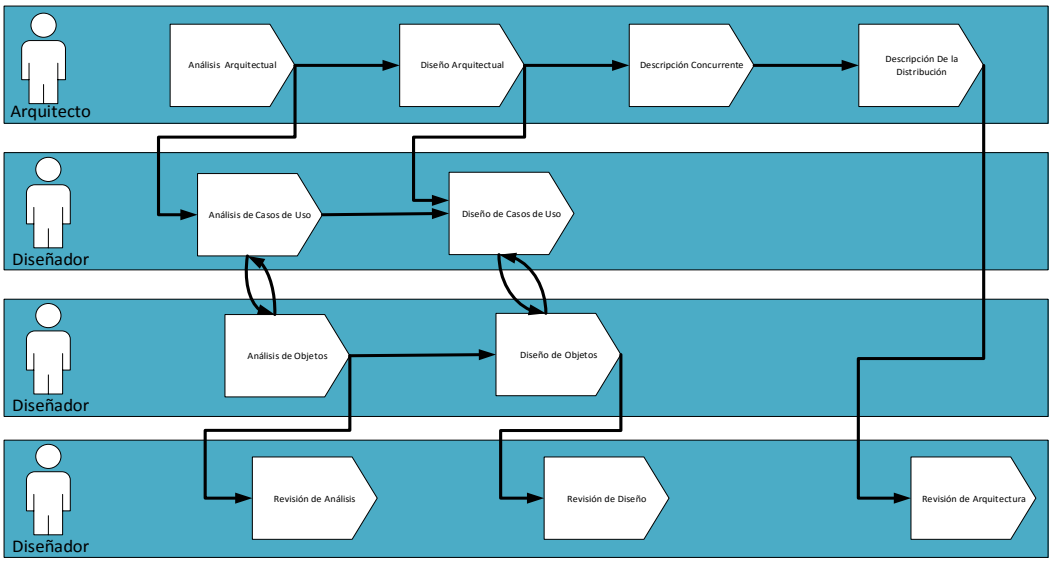

Ilustración 5: ***Ejemplo de Flujo de Trabajo manejado por la metodología RUP.***

Fuente: ***(Rational Software- The Software Development Company, 1998, p. 9)***

A continuación, se detalla una descripción de los flujos de trabajo esenciales planteados por RUP, graficados en la *Ilustración 6*, los cuales se los considera como los necesarios para llevar un proceso de calidad.

- **FLUJOS DE TRABAJO BÁSICOS**

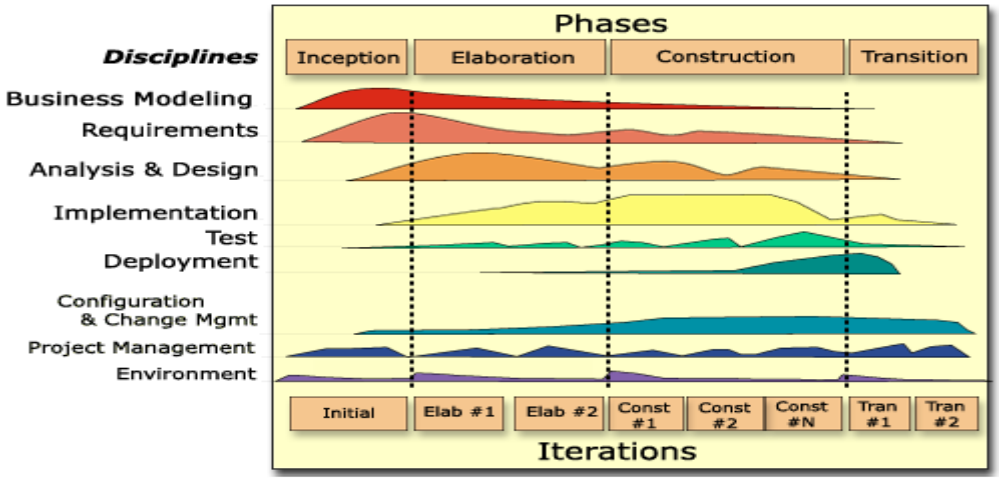

Ilustración 6: ***Nueve Flujos de Trabajo Básicos de RUP.***

Fuente: ***(Rational Software- The Software Development Company, 1998)***

1. **Modelado del Negocio:** RUP plantea una clara interacción entre la Ingeniería del Software y el Modelado del Negocio para que los requerimientos sean planteados y claramente entendidos. Esto haciendo uso de un lenguaje unificado como lo es UML, mediante el cual se generan documentación de los procesos del negocio usando Modelos de Caso de Uso del Negocio. Con esto se busca establecer los procesos del negocio de la organización para que los mismos se puedan analizar e interpretar sobre lo que se deba trabajar. (Rational Software- The Software Development Company, 1998, p. 10)

2. **Requerimientos:** En este flujo de trabajo se busca establecer *qué* debe hacer el sistema, esto con el fin de acordar con el cliente la descripción de los requerimientos esperados. De igual manera se determinan los actores de cada fase del proceso, se plantean los Casos de Uso y la descripción de la misma.

Para hacer evidente los acuerdos establecidos sobre los requerimientos del cliente, se generan documentos tangibles que detallen los requerimientos establecidos, además de describir paso a paso cada uno de los casos de uno necesarios del sistema, sus interacciones y responsables. De igual manera se detallan los requerimientos No funcionales del sistema. (Rational Software- The Software Development Company, 1998, p. 11)

3. **Análisis y Diseño:** La meta del flujo de trabajo de Análisis y Diseño es evidenciar el como él sistema va a ser desarrollado durante la fase de implementación. Durante esta fase se infiere el cómo se realizan las distintas actividades y funcionalidades del sistema, todo esto basándose en el análisis de los casos de uso desarrollados en la fase de Requerimientos.

4. Como resultado se obtiene un Modelo de Diseño y un Modelo de Análisis. El Modelo de Diseño consiste en el diseño y estructura de las clases que va a manejar el sistema, de este modelo se obtiene el cómo las distintas clases van a colaborar entre sí para cumplir con los casos de uso. (Rational Software- The Software Development Company, 1998, p. 11)

5. **Implementación:** En la fase de implementación como su nombre lo dice, su objetivo es definir la organización de código y la implementación de las clases en términos de componentes del sistema, integrar los componentes del sistema con el fin de desarrollar las interacciones de los mismos. Y un punto importante son las pruebas unitarias que se deben realizar a los distintos componentes para validar el cumplimiento de los mismos.

   El desarrollo del sistema se lo efectúa a través de la implementación de componentes. RUP describe cómo se puede reusar componentes existentes o se debe implementar nuevos componentes definiendo responsabilidades para que el sistema en sí sea mantenible y aumentar las posibilidades de reuso del mismo. *(Rational Software- The Software Development Company, 1998, p. 12)*

6. **Pruebas:** El Flujo de Pruebas dentro de RUP, es un proceso que se lo realiza a lo largo de todo el proyecto, esto con el fin de detectar tempranamente defectos y mitigar potenciales riesgos que se puedan presentar y así reducir el costo de arreglar los defectos.

   Las pruebas se las realice bajo tres dimensiones: funcionalidad, rendimiento de la aplicación y rendimiento del sistema, donde por cada una de estas dimensiones de calidad el proceso de pruebas describe como se deben llevar las mismas a través del ciclo de vida de pruebas, Planificación, Diseño, Implementación, Ejecución y evaluaciones de las pruebas. (Rational Software- The Software Development Company, 1998, p. 12).

   RUP pone en evidencia que el proceso de pruebas se lo debe manejar desde el inicio del proyecto, paralelo al ciclo de vida de desarrollo del software se debe llevar un ciclo de pruebas que permita realizar una verificación temprana de integración e interacción de los componentes del sistema sea adecuadamente igualmente verificar que los requerimientos se hallan implementado adecuadamente. Y más importante aún, detectar tempranamente los defectos y defectos potenciales cuando aún el costo de reparación y mitigación es bajo.

7. **Despliegue:** Una vez completado el Flujo de Pruebas el Flujo de Despliegue busca principalmente el producir exitosamente un producto entregable al usuario final, Entre las actividades que se cumplen durante Flujo de Despliegue destacaremos las referentes a las pruebas, Como es el caso de la planificación y ejecución de pruebas a nivel del entorno del usuario y del análisis del usuario en cuanto al cumplimiento de los requerimientos.
   Las Pruebas beta, son las pruebas que se realizarán en un entorno real, bajo condiciones reales de uso y efectuadas por el usuario final del sistema, quien brindará los comentarios necesarios que el considere para acomodar las funcionalidades del sistema a las necesidades que el usuario haya expresado en el Flujo de Requerimientos. (Rational Software- The Software Development Company, 1998, p. 13)

8. **Gestión del Proyecto:** La Gestión de Proyectos se interpreta como el manejar el cumplimiento de los requerimientos, mitigar riesgos, sobrellevar contratiempos con el fin de entregar exitosamente un producto al usuario final que cumpla con sus requerimientos iniciales y satisfaga las necesidades que este tenga. (Rational Software- The Software Development Company, 1998, p. 13)

9. **Gestión de Cambios y Configuración:** Durante el Flujo de Gestión de Cambios y Configuración, se busca gestionar los artefactos generados durante todo el desarrollo, para que los mismos ordenadamente sirvan de respaldo y base para la constante evolución de los requerimientos y especificaciones del sistema, al igual que los cambios de los artefactos por las distintas actividades y los distintos actores. (Rational Software- The Software Development Company, 1998, p. 13)

10. **Entorno:** El Flujo de Entorno busca manejar y proveer el entorno de desarrollo al igual que las herramientas necesarias para el desarrollo eficiente del ciclo de vida de desarrollo del software y paralelamente a este el proceso de pruebas y control de calidad del software. (Rational Software- The Software Development Company, 1998, p. 14)

A diferencia del modelo de desarrollo de Software Ágil Scrum, anteriormente analizado, RUP plantea fases claras y definidas, las cuales tienen objetivos, estrategias y pasos a seguir para su cumplimiento. Todo esto con la mentalidad de entregar un proyecto meticuloso y de calidad.

Es así que, en el quinto flujo de trabajo, planteado por RUP, el modelo define que la verificación de la calidad debe llevarse a la par con el desarrollo del producto de software ya que la calidad se la debe evaluar frente a los requerimientos iniciales basándose en la confiabilidad, funcionalidad y rendimiento de la aplicación y del sistema en sí. Para ello RUP asiste en la planificación, diseño, implementación, ejecución de pruebas y evaluación de resultados con el fin de llevar un control de la calidad del software.

RUP, al ser una metodología tradicional, hace uso de mucha documentación, la cual respalda el progreso y objetivos alcanzados durante el ciclo de vida de desarrollo de software. Estos llamados Artefactos combinados con un proceso y una guía a seguir, aportan la estabilidad y fiabilidad al proceso de desarrollo, lo mismo sucede con el ciclo de vida de las pruebas, ya que permite brindar un flujo constante de introducción calidad en el proceso de desarrollo.

Como plantea RUP, el manejar un ciclo de vida de pruebas paralelo al ciclo de vida del software aporta el poder determinar, enfrentar y hasta mitigar inconvenientes y contratiempos en el desarrollo que puedan resultar en costos extras e innecesarios al proyecto.

2.2.3. *Estándares para la calidad de software*

No se puede hablar de calidad sin mencionar estándares para la calidad, los dos son términos íntimamente ligados. Por un lado, la calidad busca aportar valor extra a un producto o proceso y por el otro lado, los estándares establecen los pasos y caminos necesarios para que ese valor sea continuo y perdurable en el tiempo.

En mundo laboral, hoy en día se maneja una muy fuerte competencia entre empresas para ofertar el mejor producto al público. Es decir, ofrecer un producto de calidad que cumpla con los requerimientos y expectativas del usuario final. Este producto que se busca satisfaga las necesidades del usuario se espera que sea de calidad, pero para que esta calidad se refleje en el producto terminado se debe buscar que la elaboración y construcción del mismo mantenga contantemente involucrado el concepto de calidad.

Con el aumento de problemas por solucionar y necesidades por satisfacer, se ha vuelto aún más laborioso el camino de encontrar la óptima solución a los requerimientos que el cliente espera que se cumplan de la mejor manera. Este intrincado camino a la satisfacción del cliente nos lleva a considerar dos frentes en la calidad, por el un lado se espera desarrollar y entregar un producto de software, confiable, a tiempo y apegado al presupuesto; el otro frente se vuelve más complicado, debido que aquí se verá la perspectiva del cliente que desea que todos sus requerimientos se cumplan a cabalidad.

Para alcanzar la calidad esperada por el cliente y el cumplimiento de manera óptima de los requerimientos por el planteados. Instituciones internacionales, líderes en el campo de la computación, como IEEE plantean estándares y modelos de calidad que establecen los pasos, controles y lineamientos que se deben seguir para alcanzar un nivel de calidad necesitado para un óptimo producto final.

En resumen, los estándares y modelos de calidad nos permiten de mejor manera brindar una óptima solución que satisfaga las necesidades del cliente, igualmente aportan para que el producto y la empresa alcancen una mayor competencia en el mercado y de igual manera encamina a la detección temprana y mitigación de riesgos, existentes y potenciales, todo esto con el principal fin de que la calidad del producto se trasforme en reéito económico para la empresa.

La una buena implementación de un sistema, no sólo involucra el seguir todos y cada uno los pasos que señalan los modelos o estándares. Es necesario igualmente tener un proceso y prácticas documentadas que se las aplique diariamente, igualmente debe haber un compromiso con la empresa y el mejoramiento continuo individual y colectivo por parte de cada uno de los integrantes de la organización.

Hoy en día hay gran número de estándares y modelos, que con distintos enfoques y desde varios frentes, buscan aportar los pasos, métodos, medios y prácticas que ayuden a la alcanzar los objetivos de calidad esperados en un producto o un proceso. Pero en sí, ¿qué son los estándares y qué son los modelos?

Partamos definiendo ambos términos, el Instituto de Administración de Proyectos, PMI, por sus siglas en inglés, el cual es un desarrollador acreditado de estándares del Instituto Nacional Americano de Estándares[24] (ANSI), define un estándar como: *Un documento establecido por consenso, aprobado por un cuerpo reconocido, y que ofrece reglas, guías o características para que se use repetidamente*. (Project Management Institure , 2016).

Por otro lado, la Asociación Española para la Calidad, AEC, define a un Modelo como: *Referencias que las organizaciones utilizan para mejorar su gestión. Los modelos, a*

[24] (American National Standards Institute, 2016)

diferencia de los estándares, no contienen requisitos que se deben cumplir sino directrices para la mejora (Asociación Española para la Calidad, 2016).

Tanto el Instituto de Administración de Proyectos (PMI) como la Asociación Española para la Calidad, plantean que los estándares y los modelos respectivamente, manejan un enfoque en la búsqueda de alcanzar la calidad de un producto, La diferencia se marca en el cómo los estándares y los modelos trabajan y gestionan los procesos, acciones y recursos para alcanzar la calidad deseada.

Los modelos plantean directrices que se deberían adoptar si se busca generar una mejora en la gestión de la calidad, sin establecer prerrequisitos que las organizaciones deban cumplir si buscan adoptar y ejecutar un modelo.

En contraste, un estándar establece las reglas, guías y características que un producto o proceso debe cumplir para asegurar la calidad. El cumplimiento de los lineamientos que plantea el estándar y la evaluación de que estos son repetidos continuamente en los ciclos de vida de desarrollo de software y aseguramiento de la calidad permite que un producto o proceso sea certificado y respaldado por el estándar.

Al ser un estándar desarrollado por instituciones internacionales, tales como la IEEE, ISO o PMI entre otras, que avalan la utilidad del mismo y que ratifican que el producto o proceso que sea guiado por el estándar cumplirá con un nivel repetible e incrementable de calidad, pone en evidencia que las normas y lineamientos consensuados, con estudios y análisis de las características y normas planteadas, respaldarán el aseguramiento de la calidad aportado por el estándar.

Referenciando al Instituto de Ingeniería Eléctrica y Electrónica (IEEE), este plantea varios estándares enfocados a la calidad entre ellos podemos mencionar al IEEE 829, el cual es un estándar que contiene documentación, pasos y actividades que se debe llevar para cumplir con un proceso de pruebas efectivo y el IEEE 1028, el cual por su lado define tipos de revisiones y auditorias de software, junto con los procesos requeridos para la ejecución de cada uno de los cinco tipos que el estándar propone.

Por otro lado, la Organización Internacional para la Estandarización (ISO), plantea sus propios estándares, de igual manera, con un enfoque hacia la calidad como son: ISO/IEC

25010:2011 antes conocida como ISO 9126 el cual establece los requisitos y la evaluación de la calidad del producto de software tanto interna como externa, manejando un enlace entre las necesidades y la evaluación. El estándar ISO/IEC 29110 en su paquete de Integración y Pruebas que se enfoca en definir los pasos, actividades y herramientas que una pequeña empresa debe utilizar para un óptimo aseguramiento de la calidad, y uno de los estándares más utilizados en el control de la calidad como lo es ISO/IEC – 15504 (SPICE)[25] *Software Process Improvement and Capability Determination*, o en español *Determinación de la Capacidad y Mejora de los Procesos de Software*. El cual es un modelo para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas informáticos y productos de software.

En los estándares descritos anteriormente, planteados tanto por ISO como por IEEE, se establecen pasos, actividades, procesos, técnicas y herramientas que una organización debe adoptar y practicar con el fin de asegurar un proceso de calidad eficaz y repetible que respalde al desarrollo de un producto de software que cumpla con los requerimientos establecidos y que satisfaga las necesidades del cliente.

2.3. CMMI – Aporte a la Mejora Continua

En la búsqueda de alcanzar un nivel de calidad alto y contante, que respalde el valor que pretendemos aportar a nuestros productos, se anhela seguir todos los pasos y realizar todas las tareas que se encuentren a nuestro alcance para conseguir dicho objetivo de entregar un producto de alto valor y calidad.

Para ayudarnos a alcanzar ese nivel de calidad deseado, se recibe un valioso aporte por parte de los Modelos de Calidad, este es el caso de CMMi (Capability Maturity Model Integration)[26], por sus siglas en inglés. El modelo de calidad CMMi que nace con el fin de aportar un conjunto de pautas y buenas prácticas que ayudan a mejorar los procesos que intervienen en el desarrollo del software, de igual manera presentar criterios que permitan determinar el nivel de madurez de una organización de desarrollo de software.

[25] (Atom - SPICE/ISO /IEC 15504, 2012)
[26] (Software Engineering Institute, 2007, p. 1)

Dicho modelo fue publicado en el año 2.001, por el Instituto de Ingeniería del Software SEI[27], tras 17 años de trabajo e investigación conjunta entre el Departamento de Defensa Americano y la Universidad Carnegie Mellon, fundadores del SEI. El modelo parte de una premisa de que la calidad de un producto es directamente relacionada con la calidad de los procesos que intervengan en la producción del mismo.

Para definir CMMi, se debe definir que se considera como *la madurez de una organización*, Juan Palacio, en su publicación *Sinopsis de los modelos SW-CMM y CMMi* plantea que, la madurez de una organización se evalúa de acuerdo a la medida en que el trabajo de la misma está siendo guiado por procesos definidos y en el grado en que estos se encuentran implementados dentro de la organización y son conocidos y aplicados por cada miembro de la organización, todo esto manejando una evaluación de los procesos con un fin de llevar una mejora continua[28]. Esto determina el mayor o menor grado de madurez que una organización haya alcanzado.

CMMi, basándose en CMM, plantea cinco niveles de madurez para organizaciones de desarrollo de productos de software[29]:

Nivel 1: Inicial

Los resultados de calidad son resultado de las herramientas y del personal, más no resultados de los procesos, ya que los mismos pueden no ser utilizados o no existen.

Nivel 2: Gestionado

Se alcanza el nivel Gestionado si la organización realiza tareas básicas de gestión de proyectos, gestión de requerimientos, control de versiones y de trabajos realizados, con ello permitir que el personal pueda emplear los resultados de éstas, aplicar en nuevos proyectos.

[27] Ibíd., p. 2.
[28] (Palacio, Juan; Sinopsis de los modelos SW-CMM y CMMi, 2006, p. 1)
[29] Ibíd., p.1-3

Nivel 3: Definido

Los procesos comunes están debidamente documentados y a la disposición de los miembros del equipo, igualmente el personal ha recibido la preparación necesaria para entender y ejecutar los procesos definidos.

Nivel 4: Gestionado Cuantitativamente

La organización mide la calidad del proceso y del producto de manera cuantitativa, apoyándose en métricas establecidas. La capacidad de los procesos empleados es previsible, y el sistema de medición permite detectar si las variaciones de capacidad exceden los rangos aceptables para adoptar medidas correctivas.

Nivel 5: Optimizado

Manejo de la mejora continua de los procesos que afectan a toda la organización, cuenta con los medios para identificar las debilidades y reforzar la prevención de defectos. Se analiza de forma sistemática datos relativos a la eficacia de los procesos para analizar el costo y el beneficio de las adaptaciones y las mejoras. Se analizan los defectos de los proyectos para determinar las causas y su mapeo sobre los procesos.

Dentro de CMMi, se maneja otro concepto muy importante para definir la calidad de un proceso, aportada por una organización, la Capacidad, atributo de un proceso, que expresa si un proceso solo se ejecuta o para este se debe realizar una planificación previa, igualmente determina si está formalmente definido y si es medido y mejorado regularmente.

Son seis niveles de capacidad planteados por CMMi, para medir la capacidad de los procesos que una organización pueda poseer.

0,- Incompleto: No se realiza el proceso o no se llegan a alcanzar los resultados esperados.

1,- Ejecutado: Se realiza el proceso y se llegan a alcanzar los resultados esperados.

2,- Gestionado: El proceso requiere una planificación y una revisión previa y el mismo se evalúa para comprobar que cumple con requerimientos previos.

3,- Definido: Además de ser un proceso gestionado, el mismo se alinea a las políticas y directivas de la organización.

4,- Cuantitativamente Gestionado: Además de ser un proceso definido, este es controlado empleando mediciones cuantitativas.

5,- Optimizado: Además de ser un proceso cuantitativamente gestionado, de forma periódica este es revisado y modificado para ajustarlo a los objetivos del negocio.

2.3.1. *TMMi – Modelo de Madurez de la Calidad de Pruebas*[30]

Con la incipiente búsqueda de alcanzar una mayor calidad de los productos de software, en un mercado muy competitivo, las empresas y organizaciones se han visto en la necesidad de realizar altas inversiones para conseguir el nivel de calidad deseado, pero alcanzar niveles altos se ha vuelto aún más complicado por el tamaño y complejidad del software que el cliente espera, quien hoy en día es aún más exigente.

Un punto de partida para alcanzar buen nivel de calidad y mejora en el proceso de desarrollo, es la aplicación del Modelo de Madurez de Capacidades. El Modelo de Madurez de Capacidad (CMM) y su sucesor el Modelo de Madurez de Capacidades de Integración (CMMi). A pesar de que las pruebas a menudo representan al menos el 30-40% de los costos totales del proyecto, se presta una atención limitada a las pruebas en los diversos modelos de mejora de procesos de software, como el CMM y CMMI.

Como respuesta, la *TMMi Foundation*[31], una comunidad creada con el fin de desarrollar un complemento para la mejora continua del proceso de pruebas y las actividades relacionadas con el control y aseguramiento de la calidad del software ha desarrollado sus propios modelos de mejora para pruebas, Integración de un Modelo de Madurez de Pruebas (TMMi). El TMMi es un modelo detallado para la mejora de procesos de prueba que trabaja de manera complementaria a CMMi.

[30] (TMMi Foundation, 2012, p. 6)
[31] Ibíd., p.6.

La Integración del Modelo de Madurez de Pruebas (TMMi) es un marco de orientación y de referencia para la mejora del proceso de pruebas de software, que abarca en un amplio a todas las actividades relacionadas con la calidad del producto de software.

Al igual que el CMMi, TMMi también maneja el concepto de niveles de madurez para la evaluación del proceso y la mejora, de igual manera identifican áreas de procesos, objetivos y prácticas que bajo los criterios de madurez de TMMi mejorarán el proceso de prueba y apoyan a lograr un impacto positivo en la calidad del producto, la productividad de ingeniería de pruebas, y el esfuerzo y tiempo empleado en el ciclo de desarrollo de las pruebas.

El fin del desarrollo de TMMi desde su creación ha sido ayudar a las organizaciones a evaluar y mejorar su proceso de prueba. TMMi busca que un proceso de pruebas mal definido, carente de lineamientos, recursos y hasta carente de personal preparado evolucione a un proceso maduro y controlado que mantenga como objetivo la prevención de defectos, donde la experiencia práctica sea un apoyo positivo al proceso. TMMi apoya establecer un más eficaz y eficiente proceso de prueba. Las pruebas se convierten en una profesión y una parte totalmente integrada del proceso de desarrollo donde se cambia el foco de realizar pruebas de detección de defectos a realizar actividades de prevención de defectos.

El modelo de madurez de pruebas (TMMi) fue concebido con la visión de atender a todos los niveles de pruebas que puedan presentar una organización, tanto pruebas estáticas como dinámicas y aspectos estructurales de las pruebas. El modelo alcanza niveles de pruebas bajos como pruebas de componentes y pruebas de integración; y niveles de pruebas altos como pruebas de sistemas y pruebas de aceptación, todas las pruebas en los distintos niveles están dentro del alcance de TMMi.

De igual manera que CMMi, TMMi está contemplado para apoyar en actividades y procesos de pruebas. Para ello plantea cinco niveles de madurez para el modelo de pruebas. Los niveles brevemente descritos en la *Ilustración 7,* reflejan las etapas por las cuales una organización debe llevar su proceso de pruebas para pasar de un proceso poco definido o escaso, a un proceso manejado, definido y optimizado.

Ilustración 7: ***Niveles de Madures de TMMi y áreas de procesos***

Fuente: **(TMMi Foundation, 2012, p. 9)**

Cada uno de los niveles de madurez de TMMi contiene un conjunto de áreas de procesos que la organización debe implementar para alcanzar la madurez de su proceso en ese nivel. La organización conseguirá alcanzar la madurez de su proceso de pruebas, enfocándose en un grupo determinado de actividades que le permitan mejorar y aprender en el proceso de mejora, para establecer los cimientos para los siguientes niveles.

Se debe identificar que los esfuerzos en mejorar deben estar enfocados en las necesidades de la organización bajo el contexto del ambiente del negocio.

Nivel 1 Inicial:

* Proceso caótico y no definido.
* No se tiene un ambiente estable que soporte el proceso de pruebas.
* El éxito del proceso de pruebas depende de las habilidades individuales de los probadores
* El objetivo de las pruebas no es más que determinar que el producto de software funcione sin mayores fallos.

* Se libera el software sin una revisión consistente del nivel de calidad y riesgos potenciales presentes.
* El producto difícilmente satisface las necesidades del cliente, no es estable.
* Se carece de recursos, herramientas y preparación de los probadores.
* No se puede repetir el éxito en proyectos posteriores, y la calidad del producto no es la esperada.
* Se libera el proyecto fuera del tiempo estimado y se sobresale del presupuesto.

Nivel 2 Manejado:

- El proceso de pruebas se vuelve un proceso manejable, disciplinado y perdurable en tiempos de crisis.
- Se sigue creyendo que el proceso de pruebas es un proceso post desarrollo.
- Una macro estrategia de pruebas es establecida. Se define un plan de pruebas
- Técnicas de manejo de riesgos son empleadas para identificar los riesgos del producto, basados en los requerimientos documentados.
- El plan de pruebas define que pruebas se deben aplicar, cuando y quién es el responsable de aplicarlas.
- Las pruebas son controladas y monitoreadas con el fin de saber si se ejecutan de acuerdo al plan de pruebas y poder tomar acciones correctivas si fuera el caso.
- Es evidenciable el estado del proceso de pruebas.
- Se emplea técnicas de diseño para elaborar los casos de pruebas necesarios para el proceso.
- El proceso de pruebas puede comenzar tardíamente en relación con el proceso de desarrollo.
- El objetivo principal de las pruebas en este nivel es asegurar la satisfacción de los requerimientos.
- Los defectos encontrados en este nivel son causa del inicio demorado de las pruebas.

Nivel 3 Definido:

- El proceso de pruebas ya no es una fase que sigue al proceso de desarrollo, se maneja como una actividad paralela y complementaria al proceso de desarrollo.
- Se realiza una planificación temprana de pruebas a ejecutar y se documenta debidamente la planificación.
- Se establece un proceso estándar de pruebas, es la base para el proceso de pruebas en el nivel 3 de madurez.
- Se desarrolla el plan maestro de pruebas a ejecutarse durante todo el ciclo de vida.

Nivel 4 Medido:

- La infraestructura formada por la madurez alcanzada en los niveles 2 y 3 permite que el proceso de pruebas pueda ser medible que crezca a futuro.
- Un programa de medición de pruebas, efectivo para toda la empresa, es puesto en práctica y permite la medición y monitoreo de la calidad del proceso de pruebas.
- Métricas son incorporadas al programa de medición de pruebas para apoyar a la toma de decisiones presentes y futuras.
- El programa de medición permite establecer un proceso de evaluación de la calidad en cuando a necesidades, atributos y métricas de calidad.
- El producto se lo evalúa bajo criterios cuantitativos y atributos cualificativos como fiabilidad, usabilidad y mantenibilidad.
- Revisiones e inspecciones son consideradas como parte del proceso de pruebas.

Nivel 5 Optimizado:

- La organización es capaz de lograr una mejora continua, basándose en un control estadístico del proceso de pruebas.

- Las técnicas y métodos de pruebas son optimizados y se tiene una visión continua de mejorar los procesos.
- El proceso optimizado es manejable, definido, medible, eficiente y efectivo.
- Proceso estadísticamente predecible y controlado. Enfocado en la detección de defectos.
- Un grupo para la implementación de un proceso de mejora del proceso de pruebas es conformado por personal debidamente entrenado.
- El proceso de prevención de defectos busca identificar causas de defectos para aplicar las correspondientes acciones correctivas.

2.4. Estándares Internacionales de Calidad para el Desarrollo del Software.

Dentro del mundo de la informática y la computación, una gran parte de la misma es el desarrollo de software, software que hoy en día tiene que cumplir grandes exigencias y debe alcanzar altos niveles de calidad para satisfacer las necesidades de quienes lo requieren.

Para fijar las reglas en la competencia en mercados internacionales se crearon y establecieron los estándares internacionales de calidad, los cuales son reglamentos y criterios establecidos por organizaciones avaladas internacionalmente con el fin de ayudar a definir la conformidad entre fronteras internacionales.

Las reglas y normativas estipuladas por los estándares, facilitan la realización y competitividad de las empresas y sus negocios en ámbitos globales, de igual manera permite una mayor colaboración al superarse las limitaciones regionales y globales. El enfoque de los estándares internacionales está en definir los términos en que las organizaciones compitan con sus iguales en una arena global y definir los procesos que se deban seguir para cumplir con un control de calidad[32].

Para el presente proyecto de disertación se contemplarán ciertos estándares internacionales publicados por ISO y por IEEE, los cuales se ajustan o contienen lineamientos, parámetros y/o herramientas que permitirán definir un marco de referencia de calidad que aporte a una metodología ágil.

[32] (Das, Ranjna, 2015)

2.4.1. *ISO/IEC 15504 (SPICE)*

Para el año de 1993, la comisión ISO/IEC aprobó un programa de trabajo para el desarrollo de un modelo que fuera la base de un futuro estándar internacional para la evaluación de los procesos del ciclo de vida del software. El trabajo recibió el nombre de proyecto SPICE (Software Process Improvement and Capability Determination)[33].

Para el año de 1995 se publica la primera versión del borrador del estándar, desde ahí se ha venido trabajando sobre el mismo, hasta su publicación final en 2004[34].

El estándar ISO/IEC 15504, fue concebido con el fin de aportar un marco y los requisitos necesarios para establecer un proceso de evaluación de procesos, el mismo que pueda evaluar, como ya se dijo procesos y de igual manera permita evaluar organizaciones. En este estándar se definen las competencias de un evaluador de procesos que comprenden la evaluación de proceso en sí, mejora de procesos, determinación de capacidades.

SPICE provee un conjunto de documentos, que pueden ser usados como un marco de referencia para la evaluación de procesos de software. Esta documentación puede ser utilizada por las organizaciones en fases como: planificación, manejo, monitoreo, control y mejora entre otras[35].

La evaluación del proceso de software examina los procesos para determinar donde son efectivos y alcanzan sus metas, que se ha hecho determinando la capacidad del proceso. Este acercamiento a la evaluación del proceso permite que se determine la capacidad de un determinado proceso para cumplir un requerimiento o si el mismo necesita apoyo para alcanzar la capacidad de cumplir los requerimientos. Podemos ver esto de forma gráfica en la Ilustración

[33] (Paez Bernal, 2009)
[34] (Pyhäjärvi, 2004)
[35] Ibíd.

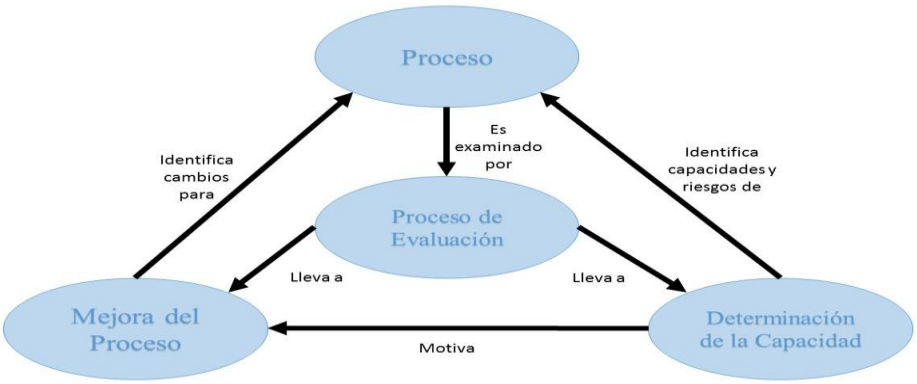

Ilustración 8: ***Proceso de Evaluación de un proceso***

Fuente: (**Pyhäjärvi**, **2004**)

2.4.2. *ISO/IEC 25010:2011*[36]

El estándar ISO/IEC 25010:2011, determina las características de calidad que se deben considerar al momento de evaluar las propiedades es de un producto de software.

La calidad del producto software se puede interpretar como el grado en que dicho producto satisface los requisitos de sus usuarios aportando de esta manera un valor. Son precisamente estos requisitos (funcionalidad, rendimiento, seguridad, mantenibilidad, etc.) los que se encuentran representados en el modelo de calidad, el cual categoriza la calidad del producto.

El modelo de calidad del producto definido por la ISO/IEC 25010 se encuentra compuesto por las ocho características de calidad que se muestran en la siguiente figura:

[36] (ISO 25000, 2015)

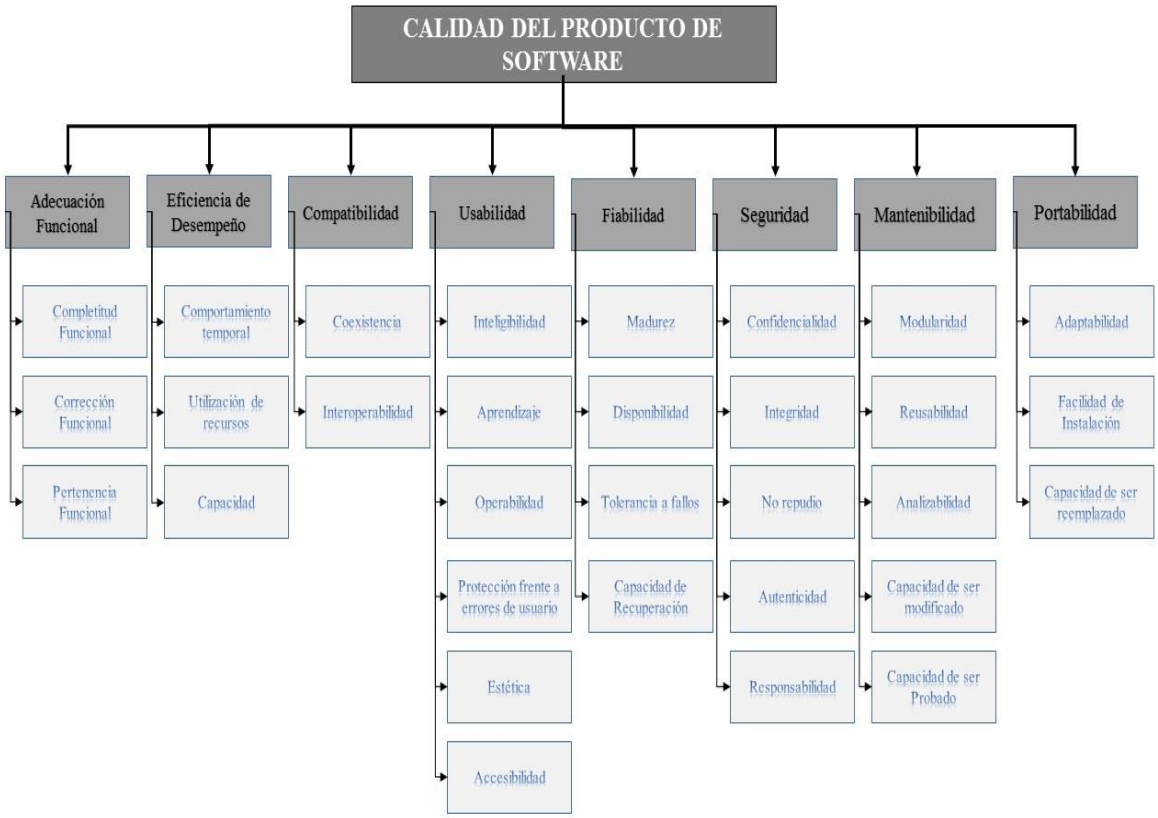

Ilustración 9: ***Modelo de calidad del producto definido por la ISO/IEC 25010***

Fuente: **(ISO 25000, 2015)**

A continuación, se describen las características de calidad, planteadas por el estándar ISO/IEC 25010:2011 para el modelo de calidad de un producto de software.

1. **Adecuación Funcional:** Representa la capacidad del producto software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas.
   - **Completitud funcional.** Grado en el cual el conjunto de funcionalidades cubre todas las tareas y los objetivos del usuario especificados.
   - **Corrección funcional.** Capacidad del producto o sistema para proveer resultados correctos con el nivel de precisión requerido.
   - **Pertinencia funcional.** Capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.

2. **Eficiencia de desempeño:** Característica que representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones.

- **Comportamiento temporal.** Los tiempos de respuesta y procesamiento de un sistema bajo condiciones determinadas.
- **Utilización de recursos.** Las cantidades y tipos de recursos utilizados cuando el software lleva a cabo su función bajo condiciones determinadas.
- **Capacidad.** Grado en que los límites máximos de un parámetro de un producto o sistema software cumplen con los requisitos.

3. **Compatibilidad**: Capacidad de dos o más sistemas o componentes para intercambiar información y/o llevar a cabo sus funciones requeridas cuando comparten el mismo entorno de hardware o software.
   - **Coexistencia**. Capacidad del producto para coexistir con otro software independiente, en un entorno común, compartiendo recursos comunes sin detrimento.
   - **Interoperabilidad**. Capacidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada.

4. **Usabilidad**: Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones.
   - **Inteligibilidad**. Capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.
   - **Aprendizaje**. Capacidad del producto que permite al usuario aprender su aplicación.
   - **Operabilidad**. Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.
   - **Protección contra errores de usuario**. Capacidad del sistema para proteger a los usuarios de hacer errores.
   - **Estética**. Capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario.
   - **Accesibilidad**. Capacidad del producto que permite que sea utilizado por usuarios con determinadas características y discapacidades.

5. **Fiabilidad**: Capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y período de tiempo determinados.
   - **Madurez**. Capacidad del sistema para satisfacer las necesidades de fiabilidad en condiciones normales.
   - **Disponibilidad**. Capacidad del sistema o componente de estar operativo y accesible para su uso cuando se requiere.
   - **Tolerancia a fallos**. Capacidad del sistema o componente para operar según lo previsto en presencia de fallos de hardware o software.
   - **Capacidad de recuperación**. Capacidad del producto software para recuperar los datos directamente afectados y reestablecer el estado deseado del sistema en caso de interrupción o fallo.

6. **Seguridad**: Capacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos.
   - **Confidencialidad**. Capacidad de protección contra el acceso de datos e información no autorizados, ya sea accidental o deliberadamente.
   - **Integridad**. Capacidad del sistema o componente para prevenir accesos o modificaciones no autorizados a datos o programas de ordenador.
   - **No repudio**. Capacidad de demostrar las acciones o eventos que han tenido lugar, de manera que dichas acciones o eventos no puedan ser repudiados posteriormente.
   - **Responsabilidad**. Capacidad de rastrear de forma inequívoca las acciones de una entidad.
   - **Autenticidad**. Capacidad de demostrar la identidad de un sujeto o un recurso.

7. **Mantenibilidad**: Esta característica representa la capacidad del producto de software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas.
   - **Modularidad**. Capacidad de un sistema o programa de ordenador que permite que un cambio en un componente tenga un impacto mínimo en los demás.

- **Reusabilidad**. Capacidad de un activo que permite que sea utilizado en más de un sistema software o en la construcción de otros activos.
- **Analizabilidad**. Facilidad con la que se puede evaluar el impacto de un determinado cambio sobre el resto del software, diagnosticar las deficiencias o causas de fallos en el software, o identificar las partes a modificar.
- **Capacidad para ser modificado**. Capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar el desempeño.
- **Capacidad para ser probado**. Facilidad con la que se pueden establecer criterios de prueba para un sistema o componente y con la que se pueden llevar a cabo las pruebas para determinar si se cumplen dichos criterios.

8. **Portabilidad**: Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno hardware, software, operacional o de utilización a otro.
   - **Adaptabilidad**. Capacidad del producto que le permite ser adaptado de forma efectiva y eficiente a diferentes entornos determinados de hardware, software, operacionales o de uso.
   - **Facilidad de Instalación**. Facilidad con la que el producto se puede instalar y/o desinstalar de forma exitosa en un determinado entorno.
   - **Capacidad para ser reemplazado**. Capacidad del producto para ser utilizado en lugar de otro producto software determinado con el mismo propósito y en el mismo entorno.

2.4.3. *ISO/IEC 29110*

El estándar ISO/IEC 29110 fue desarrollado como un estándar internacional con el fin de crear un marco de referencia para pequeñas empresas y permitir que las mismas, entren a competir en el mercado internacional. El estándar ISO/IEC 29110 se conforma de varios paquetes, los cuales definen procesos, actividades, tareas, pasos, roles productos y artefactos, necesarios que una organización deba cumplir para cumplir con un determinado paquete.

Dicho estándar consta de nueve paquetes de despliegue que soportan el perfil básico. Para el presente proyecto de disertación nos enfocaremos en el paquete de despliegue – Integración y Pruebas, este paquete cubre las actividades relacionadas a las pruebas de software de ISO/IEC TR 29110 Parte 5-1-2:2011 para pequeñas organizaciones[37].

Para manejar el paquete de despliegue – Integración y Pruebas se emplean dos procesos que son Gestión del Proyecto e Implementación de Software, los cuales están directamente relacionados con el tema, estos procesos describen actividades, tareas y roles, A continuación, se describirán brevemente los procesos, actividades, tareas y roles que trabajan bajo el paquete de despliegue – Integración y Pruebas.

- **Proceso**: Gestión del Proyecto
  * **Actividad**: Planificación del Proyecto
  * **Tareas**: Identificar las Tareas específicas a realizar para producir los Entregables y sus Componentes de Software identificados en el Enunciado de Trabajo
  * **Roles**: Gestor de Proyecto, Líder Técnico.

- **Proceso**: Implementación de Software
  * **Actividad**: Arquitectura y Diseño detallado del Software
  * **Roles**: Diseñador, Analista, Líder Técnico.
  * **Tareas**:
    - Establecer o actualizar los Casos de Prueba y Procedimientos de Prueba para pruebas de integración basadas en la Especificación de Requerimientos y el Diseño de Software,
    - Verificar y obtener la aprobación de los Casos de Prueba y Procedimientos de Prueba.

[37] (Gómez Arenas & Ramos, 2013, p. 7)

- Actualizar el Registro de Trazabilidad incorporando los Casos de Prueba y los Procedimientos de Prueba.
- Incorporar el Diseño de Software, y el Registro de Trazabilidad a la Configuración de Software como parte de la línea base.

* **Actividad**: Integración y pruebas de Software.
* **Roles**: Cliente, Probador, Programador.
* **Tareas**:
  - Entender los Casos de Prueba y Procedimientos de Pruebas.
  - Integrar el Software usando los Componentes de Software y actualizar los Casos de Prueba y Procedimientos de Prueba para las pruebas de integración, conforme sea necesario.
  - Realizar pruebas de Software usando Caso de Pruebas y Procedimientos de Prueba para la integración y documentar los resultados en el Reporte de Pruebas.
  - Corregir los defectos encontrados y realizar una prueba de regresión hasta satisfacer el criterio de finalización

2.4.4. *IEEE/EIA 1028*

El estándar 1028 planteado por IEEE en el año 2008, está enfocado en proveer los requerimientos mínimos para una revisión sistemática de un producto de software. Esto promoviendo las actividades de participación que el equipo de trabajo debe ejecutar, la documentación que se debe generar como resultado de las revisiones y documentar los procesamientos necesarios para conducir una revisión.

IEEE 1028-2008 define como llevar una revisión, planteando cinco tipos de revisiones, Revisiones Administrada, Revisiones Técnicas, Inspecciones, Revisiones Paso a Paso y Auditorias. En el estándar se describe los procedimientos para ejecutar cada una de estos tipos de revisiones.

La aplicación del estándar está contemplada para todo el ciclo de vida del software sin importar el alcance del mismo, donde se obtendría un mayor beneficio si las revisiones se las planifica desde el inicio del ciclo de vida. El estándar es aplicable a un sistema total o a un componente de un gran sistema, y esto involucrando tanto a personal interno como externos como sería el caso del cliente.

El estándar IEEE 1028:2008 maneja una estructura de siete pasos que van desde la Introducción, donde se describen los objetivos y una visión genérica de los tipos de revisiones que se deben ejecutar, los responsables y responsabilidades que la revisión conllevará, los datos de entrada necesarios para desarrollar la revisión, pasando a los criterios de entrada que son los criterios que se deben cumplir antes de iniciar las pruebas, los procedimientos que se van a ejecutar en la revisión, y para terminar deben cumplir los criterios de salida para dar por cumplida la revisión y los resultados de salida.

Acorde al estándar, este plantea un cuadro de las diferencias entre los distintos tipos de revisiones, las diferencias están representadas en la *Ilustración 10.*

	Revisión Administrada	**Revisión Técnica**	**Inspección**	**Revisión Paso a Paso**	**Auditoria**
Objetivo	Asegurar el Proceso	Evaluar conformidad con la especificación	Encontrar Incidencias	Encontrar Incidencias Examinar/Mejorar	Evaluar el cumplimiento con el estándar
Número de Miembros	Ilimitado		3-6	2-7	1-5
Tamaño de Material	Moderado a Alto		Relativamente bajo		Moderado a Alto
Líder	Administrador	Ingeniero Principal	Facilitador Capacitado	Facilitador o Autor	Auditor Principal
¿Administrador Presente?	Si	Opcional	No	No	Si
Volumen del Material	Moderado a Alto	Moderado a Alto	Bajo	Bajo	Moderado a Alto
¿Listas de Verificación?	No	No	Si	No	Si
Lista de Salida	Reporte de Administración	Reporte Técnico	Lista de Defectos	Reporte	Reporte de Auditoria

Ilustración 10: ***Diferencias entre los tipos de Revisiones del Estándar IEEE 1028:2008***

Fuente: ***(Software & Systems Engineering Standards Committee, 2008)***

2.4.5. *IEEE/EIA 829*

El estándar 829 planteado por IEEE tiene un enfoque a los procesos de pruebas de software, planteando las actividades y tareas que se deben considerar al momento de evaluar el funcionamiento adecuado del software y sus relaciones con otros componentes del sistema, para ello plantea el examinar atributos del software como son la integridad del mismo, consistencia de acuerdo a las especificaciones y capacidad de prueba que tenga para determinar su funcionabilidad. La determinación de estas características entre otras, dan la validez al proceso de pruebas y la documentación de ensayo respaldan el proceso realizado.

El propósito del estándar está en establecer un marco de pruebas común al igual que las actividades y tareas que den soporte al proceso de pruebas a lo largo de todo el ciclo de vida del software. Por otro lado, plantea las tareas de pruebas que se deban ejecutar y los requerimientos de entrada y salida necesarios para realizar un proceso de pruebas eficiente[38].

El estándar IEEE 829 pretende identificar las pruebas mínimas recomendadas para un proceso de pruebas básico, donde se defina el uso y el contenido de un Plan de Pruebas Master al igual que otra documentación que aporte al proceso de pruebas como sería documentos de Diseño de Pruebas, Casos de Prueba y de Proceso de prueba, entre otros que sean evidencia de la ejecución y de los resultados de un proceso de pruebas[39].

Este estándar está organizado en cláusulas que esquematizan el proceso de pruebas que propone, estas cláusulas están planteadas desde la Cláusula 1 a la Cláusula 17[40].

- **Cláusula 1** contiene material útil para comprender y utilizar esta norma.
- **Cláusula 2** listas de referencias normativas.
- **Cláusula 3** proporciona definiciones de términos, abreviaturas y convenciones.

[38] (Software & Systems Engineering Standards Committee, 2008, p. 2)
[39] Ibíd., p. 3.
[40] Ibíd.

- **Cláusula 4** explica el concepto de la utilización de niveles de integridad de software para determinar el alcance y rigor de los procesos de prueba.
- **Cláusula 5** describe cada proceso del ciclo de vida del software principal (usando un ciclo de vida elegido para fines ilustrativos) y enumera las actividades de prueba y las tareas asociadas con el proceso de ciclo de vida.
- **Cláusula 6** define el proceso para la elección de los contenidos de la documentación de prueba.
- **Cláusula 7** define el verbo "dirección" y requiere que cada posible documentación contenga temas que puedan ser considerado para su inclusión en la documentación de las pruebas.
- **Cláusula 8** define el contenido recomendado de un Plan de Pruebas Maestro.
- **Cláusula 9** define el contenido recomendado de un Plan o Planes de Prueba de Nivel.
- **Cláusula 10** define el contenido recomendado de un Diseño de Prueba de Nivel.
- **Cláusula 11** define el contenido recomendado de un Caso de Prueba de Nivel.
- **Cláusula 12** define el contenido recomendado de un Procedimiento de Prueba de Nivel.
- **Cláusula 13** define el contenido recomendado de un registro de la Prueba de Nivel.
- **Cláusula 14** define el contenido recomendado de un Informe de Incidencias.
- **Cláusula 15** define el contenido recomendado de una Prueba Provisional de Informe de Estado de Nivel
- **Cláusula 16** define el contenido recomendado de un Informe de Prueba de Nivel.
- **Cláusula 17** define el contenido recomendado de un Informe de Prueba Maestro.

El estándar IEEE 829 se basa en las diecisiete clausulas antes nombradas, y sobre estas cláusulas propone el proceso de pruebas con las debidas actividades, tareas, responsables y entregables que evidencien la ejecución del proceso, A continuación, en la *Ilustración 11* se evidencia un uso parcial del estándar IEEE 829 para un proceso de pruebas.

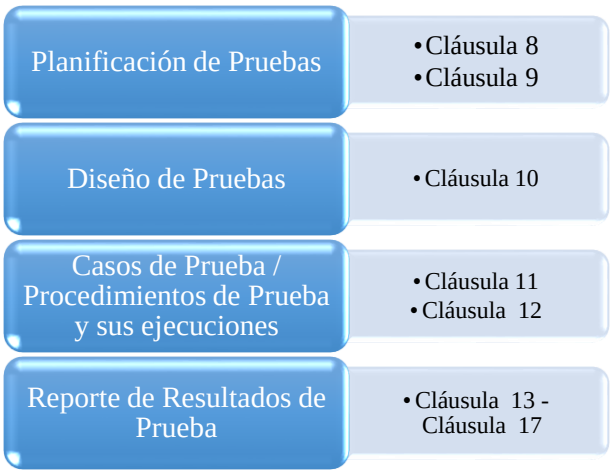

Ilustración 11: ***Uso parcial del estándar IEEE 829-2008***

Fuente: **(Software & Systems Engineering Standards Committee, 2008, p. 6)**

Capítulo. III: MARCO DE REFERENCIA – APORTE A UN PROCESO.

En el presente capítulo se establecerá qué es un Marco de Referencia y se detallará cuál es el objetivo del mismo ante un trabajo académico o de investigación. Como parte de este capítulo, se presentará de igual manera cómo un Marco de Referencia permite el conjugar múltiple información científica y de investigación para armar un sustento teórico que respalde a una propuesta de mejora.

3.1. Qué es un Marco de Referencia.

Qué se podría entender por 'Marco de Referencia', por la interpretación de los términos, se puede entender por 'Marco de Referencia' a un conglomerado de ideas y principios que nos permitan determinar un punto de partida o haga alusión a un tema en específico.

De Acuerdo al CEPPE (Comité de Evaluación de Programas de Pedagogía y Educación de México), definen a un Marco de Referencia como: "*El documento que establece los lineamientos técnico-metodológicos para realizar el proceso de acreditación del programa educativo. En él se establecen los documentos necesarios para el proceso de acreditación, los atributos a evaluar a través de los indicadores y la definición de cada una de las etapas del proceso*" (CEPPE, n.d.).

Interpretando esta definición, para el propósito del presente proyecto de disertación, podríamos definir a un Marco de Referencia como una recopilación de información argumentativa, acuerdos, parámetros, y lineamientos, manifestados con el objetivo de establecer una posición frente a una evaluación y con el fin de definir una guía para un proceso, describiendo cada uno de los pasos que se compongan seguir para el cumplimiento del mismo.

En otras palabras, un Marco de Referencia nos ofrece los lineamientos teóricos bajo los cuales la explicación, observación y análisis de la realidad y el estado actual de un proceso, se lo puede comprender con el objetivo primordial de proponer una mejora, bajo las consideraciones necesarias determinadas por el estudio realizado.

En la *Ilustración 12*, podemos tener una idea más clara de la intención de un Marco de Referencia:

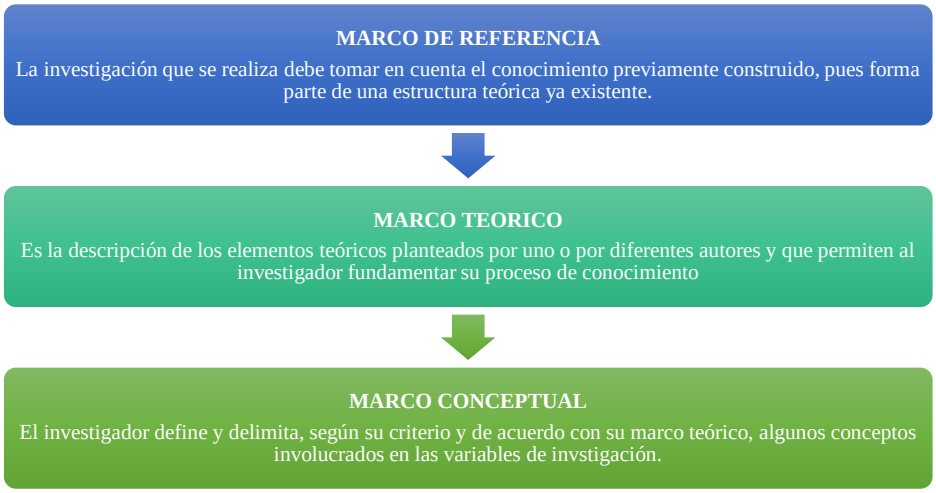

Ilustración 12: ***Esquema de la estructura de un Marco de Referencia***

Fuente: ***(Jiménez, 2012)***

3.2. Aporte teórico de un Marco de Referencia.

El aporte teórico dentro de un Marco de Referencia, de acuerdo a lo antes mencionado, se encuentra dado por el Marco Teórico, que respalda al mismo desde el diseño de éste y durante su desarrollo. Es decir, toda la investigación y levantamiento de datos previo a la definición de un Marco de Referencia, y todo el análisis realizado durante el desarrollo del Marco, aportan el sustento teórico necesario para dar validez al mismo.

Permitiendo ubicar la problemática a tratar, dentro de un conjunto de teorías y postulados existentes, el Marco Teórico de un Marco de Referencia brinda los elementos de la teoría que serán, directamente, los fundamentos para la propuesta teórica del Marco de Referencia.

3.3. Aplicabilidad del Marco de Referencia al Proceso de Calidad.

Tomando en cuenta lo antes descrito sobre lo que es un Marco de Referencia, y el aporte que el mismo da a la definición de un proceso, podríamos advertir algunas bondades que la aplicación de un Marco de Referencia pudiera dar a un Proceso de Calidad.

Partiendo de que un Marco de Referencia aporta los lineamientos y respaldos teóricos, obtenidos a partir de una investigación, un estudio y un análisis previo; se puede inferir que un Proceso de Calidad que cuente con dicho respaldo y fundamentos, estaría en la capacidad de proponer mejores prácticas y definir procedimientos que optimicen el Proceso de Calidad, para así garantizar que la aplicación del Proceso de Calidad respaldado por un Marco de Referencia representaría una mejora en el proceso en sí y por consiguiente en la calidad de los productos resultante del mismo.

Analicemos qué es un Proceso de Calidad. En un sistema de producción, un Proceso de Calidad se considera como una serie de pasos paralelos a la elaboración de un producto que el mismo debe seguir desde su planeación hasta llegar al consumidor final. Los pasos dentro de un Proceso de Calidad significarían, los procesos de pruebas y controles que el producto deba cumplir y satisfacer con el fin de alcanzar un nivel de calidad fijado por la empresa.

En el desarrollo de software de igual manera se puede considerar varios pasos dentro de un Proceso de Calidad, estos pasos de la misma forma, consisten en controles de niveles de calidad, satisfacción de requerimientos, cumplimiento de estándares y normas, entre otros, con el fin de que el producto final de software sea garantizado y respaldado por un proceso que haya aportado calidad al producto en cada fase de su elaboración.

Tomando las dos partes, un Marco de Referencia que delimite el sustento teórico que sirva de apoyo a la definición de un proceso y un Proceso de Calidad que proponga buenas prácticas y mejoras con miras a la mejora continua, podemos obtener un Marco de Referencia de Calidad que apoye al Proceso de Desarrollo de la Metodología ágil Scrum.

Capítulo. IV: DESARROLLO DE LA GUÍA DEL MARCO DE REFERENCIA

En el presente capítulo se desarrolla la estructura, pasos y procedimientos considerados para generar un aporte a la metodología ágil Scrum, a fin de consolidad un proceso de pruebas dentro de su ciclo de vida. Además de proponer los procedimientos para poner en práctica la propuesta del Marco de Referencia, se plantean criterios bajo los cuales se puede buscar generar una mejora continua de los procesos y adquisición de nuevas herramientas y habilidades del equipo de trabajo a fin de consolidar una mentalidad de trabajo, mejora y aprendizaje colectivo y colaborativo.

3.1. Estándares y normativas para la Calidad en el Desarrollo de Software.

La estructura del marco de referencia de calidad se basa en TMMi, fijando un esquema para delimitar el proceso de pruebas, el cual considera: Políticas para las pruebas, Estrategias para realizar un proceso de pruebas e Indicadores del desempeño de las pruebas. La estructura del proceso de pruebas bajo TMMi, trabaja en base a niveles de madurez, desde un nivel *Manejado* donde se establecen las bases para formar un proceso para calidad, seguido de un nivel *Definido* donde se consolida el proceso. Después, un nivel *Medido* donde la reproducción del proceso es medida para validar su ejecución, cumplimiento e identificar puntos de mejora, y para terminar en un nivel *Optimizado* donde la repetición del proceso para la calidad es fiable y además hay un control continuo de la misma para así lograr una optimización del Marco de referencia de Calidad.

Para definir los componentes del proceso de calidad, el marco de referencia se apoya en el estándar ISO 29110, éste aporta para establecer las actividades, tareas, pasos, roles y productos necesarios para conducir y administrar el proceso. En adición, los estándares IEEE 1028, IEEE 829 e ISO 25010 plantean en detalle las características de los componentes para un proceso de calidad, tales como requerimientos de entradas y de salida para las pruebas, criterios de satisfacción y responsabilidades dentro del proceso para la calidad, y además las consideraciones de los componentes que se deben tomar en cuenta para cumplir los lineamientos planteados por dichos estándares.

Para la documentación necesaria, el marco de referencia de calidad de software toma lo planteado por los estándares IEEE 829 e ISO 29110 como, por ejemplo, documentos de planificación de pruebas y descripción de requerimientos, descripción de casos de prueba y listas de verificación entre otros artefactos que den respaldo al proceso de pruebas, y sirvan como punto de partida para la repetición y mejora del proceso.

Para la evaluación del marco de referencia de calidad propuesto, éste se basa en los estándares ISO 15504 e ISO 25010, los cuales plantean los pasos para una eficiente evaluación y los factores a considerar al momento de evaluar el desempeño y mejora del proceso de calidad.

A continuación, en la *Ilustración 13*, se presenta en resumen el aporte que brindarán los distintos estándares y el modelo TMMi, a la propuesta de crear y consolidar un marco de referencia que aporte al proceso de calidad dentro de una metodología ágil y apoye al desarrollo de software.

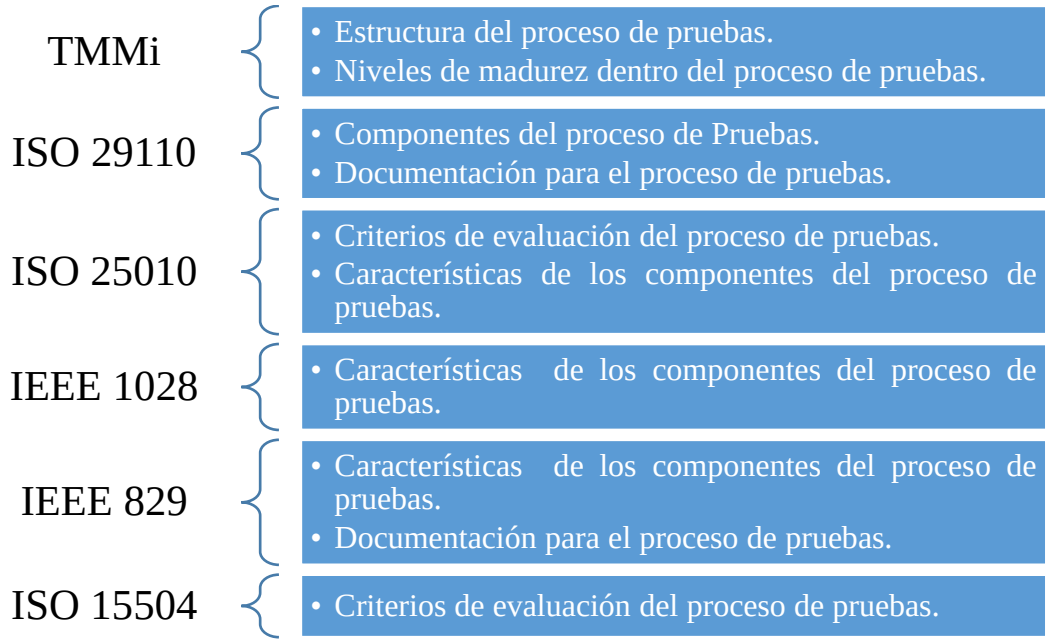

Ilustración 13: ***Modelos y Estándares empleados para la Propuesta de Marco de Referencia de Calidad***

La estructura de los niveles de madurez para el proceso de pruebas y los componentes de los mismos, necesarios dentro del marco de referencia para guiar un proceso de calidad, se plantea una organización desde un *Nivel 2* de Madurez, ya que se considera como *nivel 1*, un estado inicial de pruebas, donde la organización consta de técnicas y estrategias de pruebas basadas en los conocimientos y experiencia individual de los miembros del equipo, más no en un proceso de pruebas definido. En este nivel inicial, las pruebas aplicadas a un producto de software, generan resultados no medibles, no totalmente repetibles en futuros proyectos y aún más las pruebas no generan evidencia de actividades realizadas y con ello no se puede prevenir repetir errores pasados, además no se puede buscar una mejora en el proceso ya que no se tiene una base sobre lo cual mejorar.

Se plantea un Nivel 3 de Madurez, donde los procesos, pasos y procedimientos que se efectuaban en el Nivel 2, a lo mejor de las habilidades y conocimiento individual, ahora se los realicen de manera administrada, medible y con la posibilidad de generar una retroalimentación de la ejecución del proceso con la intención del mejoramiento del mismo.

Se seguiría con un Nivel 4 de Madurez, donde la correcta administración de los pasos, procesos y procedimientos en la ejecución de las pruebas se logre obtener datos del desempeño del proceso de pruebas que permitan determinar sus puntos débiles para solventarlos y sus puntos fuertes para incentivarlos, y con esto crear un proceso de mejora para el proceso de pruebas. Durante este nivel se deben generar una serie de artefactos y documentación que sirvan de evidencia del desarrollo de las pruebas y de base de conocimiento para futuros proyectos.

Posteriormente se llegaría a un Nivel 5 de Madurez, que como principal meta tiene la optimización. Optimización de los recursos, personal y de los procesos, manejando un proceso de mejora continua que permita la evolución y mejora del proceso de prueba.

3.2. Desarrollo de la guía del Marco de Referencia de Calidad de Software.

3.2.1. TMMi Nivel 2 – Manejado:

El Nivel 2 – Manejado, comprende un estado del proceso de pruebas donde el mismo tiene un marco de trabajo, un proceso común donde las habilidades y conocimientos individuales actúan en conjunto, para alcanzar los requerimientos funcionales del producto, esto bajo un procedimiento homologado dentro del grupo de trabajo.

En este nivel se plantean las estrategias y políticas que se debe seguir para así proceder con una ejecución planificada de las pruebas diseñadas para el producto, de esta manera se puede plantear un monitoreo y control de las pruebas a aplicar.

3.2.1.1. REQUISITOS A CUMPLIR

Dentro del Nivel Manejado, se debe cumplir con ciertos requisitos que aseguren el cumplimiento de lo planteado por el marco referencial. Estos requisitos están pensando en apoyar a la metodología ágil Scrum. Como se mencionó en el apartado 2.2.1 Metodología Ágil – ¿Qué nos dice Scrum?[41], evidenciamos que el proceso de desarrollo de Scrum está guiado por una serie de buenas prácticas y actividades que conducen dicha metodología.

Tomando los lineamientos y principios de una metodología ágil vistos en el apartado antes mencionado, se plantea que las actividades de un proceso de calidad deben seguir el mismo enfoque, donde las actividades deben ser puntuales, los resultados deben ser comunicativos y altamente entendidos por el equipo de pruebas. Con esto se definen los siguientes requisitos que se deben alcanzar con el fin de cumplir con un nivel manejado de un proceso de pruebas.

- Definir actividades para catalogar el tipo de objeto de prueba.
- Definir procedimientos para cada tipo de objeto de prueba.
- Definir los requerimientos del producto del software para la generación de los casos de prueba.
- Definir el nivel de profundidad de las pruebas a ejecutar.
- Definir un diseño de las pruebas a ejecutar en base a los requerimientos funcionales del producto.

[41] Metodología Ágil – ¿Qué nos dice Scrum?

- Definir un calendario para la ejecución de las pruebas.
- Definir procedimientos de monitoreo y control de la ejecución de las pruebas aplicadas.

3.2.1.2. TÉRMINOS A MANEJAR

- **Objeto de Prueba:** Todo componente de un producto de software o el producto completo en sí, que sea estable, funcional, completo y que pueda ser probado de acuerdo a los requerimientos del mismo, que hayan sido definidos de antemano en la definición del objeto a probar. Los componentes pueden ser una página web, una funcionalidad, un módulo de un sistema, un esquema de base de datos, una sentencia, entre otros,

- **Prioridad del Objeto de prueba:** De acuerdo a la trascendencia que tenga la evaluación del objeto de prueba frente a sus semejantes se determina la prioridad del mismo, esto igualmente está definido por el orden en que los distintos componentes del producto de software deban ser probados.

- **Criticidad del Objeto de prueba:** La criticidad de un objeto de prueba dependerá del impacto que tenga frente a otros objetos de prueba y el entorno, es decir los cambios o fallos que se deban evaluar en un objeto de prueba, cuanto influencian en el correcto funcionamiento de los demás componentes del sistema o del sistema en sí. La criticidad de un objeto de prueba puede hacer cambiar la prioridad del mismo.

- **Niveles de profundidad de las pruebas:** Los niveles de profundidad de las pruebas están ligados a la exhaustividad que se requiere con que las mismas sean ejecutadas, de acuerdo a los requerimientos y las prioridades del negocio, la criticidad del objeto de prueba ayuda a definir cuan exigentes deben ser las pruebas a ejecutar

3.2.1.3. CRITERIOS A MANEJAR

Es importante definir ciertos criterios que se deben considerar para establecer los fundamentos de un proceso de calidad. Los criterios a continuación definidos ayudan a contextualizar el proceso y delimitar las actividades que se desarrollen en la ejecución de pruebas.

- **Criterios para establecer los requerimientos funcionales del producto**: Los requerimientos del producto de software son definidos, de acuerdo a la metodología Scrum, por los Product Owner[42]. Con el fin de desarrollar casos de prueba eficientes y acordes a las exigencias del cliente, los requerimientos deben ser bien definidos, claros, no ambiguos y documentados para que sea posible comprobar su cumplimiento post evaluación del objeto de prueba.

- **Criterios para definir los tipos de objeto de prueba:** Tomando en cuenta que un objeto de prueba es todo componente de un sistema o sistema en sí, que es estable y puede ser probado, los tipos de objeto prueba se los clasifica en tres tipos a continuación descritos.

  * Nuevo objeto o componente de un objeto de prueba susceptible a ser probado.
  * Error, defecto o falla de un objeto susceptible a ser probado
  * Corrección o un cambio del objeto susceptible a ser probado debido a un cambio del requerimiento.

- **Criterio de prioridad del objeto de prueba:** La prioridad de un objeto está sujeta a la lógica del negocio y los requerimientos detrás del mismo. Si son varios objetos de prueba que son componentes de un sistema, se debe evaluar la necesidad de que un componente sea probado antes que otro tomando en cuenta la lógica del negocio y los aspectos técnicos del sistema en sí.

[42] Definición de Product Owner

Otra manera de determinar la prioridad de un objeto de prueba pudiera ser la complejidad que el mismo tenga, donde ahí quedaría a responsabilidad del probador encargado el determinar la prioridad para cada objeto a ser probado. Muchas veces la experiencia del probador es un factor importante para determinar el orden en prioridad en que el proceso de prueba se deba ejecutar de acuerdo al marco de referencia del proceso.

- **Criticidad del objeto de prueba:** Para definir la criticidad de un objeto de prueba se debe evaluar el impacto del mismo dentro del sistema y frente a otros componentes. Un alto o bajo impacto, que pueda tener un objeto de prueba, puede determinar la prioridad, la destinación de recursos y la planificación que se debe brindar a los cambios y pruebas que se ejecuten sobre dicho objeto de prueba.

- **Criterios para delimitar los niveles de profundidad de las pruebas:** Para definir el nivel de profundidad que deban tener las pruebas, se debe tomar en consideración los requerimientos que el objeto de prueba tenga, además también se debe examinar las prioridades del negocio. Un proceso de pruebas de alto nivel considera una alta inversión de personal, recursos, tiempo y dinero.

  Factores como el tiempo, el dinero, recursos técnicos y de personal son factores que influencian en determinar cuan exhaustivas deben ser las pruebas, es por ello que la metodología Scrum aporta con la continua comunicación entre las partes involucradas, así se obtienen términos claros sobre el tiempo disponible, recursos y requerimientos que determinen el nivel de las pruebas.

- **Criterios de estandarización del diseño de pruebas y casos de uso:** La estandarización del diseño de las pruebas y de los casos de uso corresponde a, que todos los diseños de los objetos de pruebas y todos los casos de uso generados de acuerdo a los requerimientos, tengan un formato homologado para que todos tengan el mismo nivel de detalle necesario para las pruebas.

Un formato homologado significa que el nivel de detalle y descripción tanto de los diseños como de los casos de prueba sea el adecuado para que las pruebas puedan ser ejecutadas de la mejor manera y competencia, una definición clara de los requerimientos ayuda a que los diseños se ajusten más a la realidad de lo esperado por el cliente y que los casos de prueba cubran todas las características esperadas que cumpla el objeto de prueba.

3.2.1.4. ARTEFACTOS A GENERAR

- Plan Maestro de Pruebas
- Esquema del Diseño de Pruebas
- Esquema de Casos de Prueba

3.2.2. TMMi Nivel 3 – Definido:

En un nivel de madurez 3, define al proceso de pruebas no como una actividad post desarrollo, sino una actividad paralela ligada al ciclo de vida de desarrollo desde el inicio del proyecto, es decir, existe la planificación de un proceso de pruebas paralelamente a la planificación del desarrollo del software.

Apoyando al proceso de la metodología de Scrum, el realizar una planificación de sprint[43] para el proceso de pruebas al inicio del proyecto, permite definir y clarificar los requerimientos del producto para poder generar modelos y casos de pruebas más eficientes que cubran de mejor manera la evaluación del cumplimiento de los mismos, la planificación de las pruebas desde el inicio del proyecto ayuda a preparar el entorno de prueba y los requerimientos del entorno que las pruebas necesitan para su ejecución óptima.

Al igual que el Equipo Scrum, un equipo de pruebas Scrum es un equipo organizado y participativo, no es un grupo aparte, el equipo de pruebas trabaja de la mano con el resto del equipo para establecer de la mejor manera los casos de prueba y las consideraciones que se deban tomar en cuenta al momento de evaluar el software. De igual manera el equipo de

[43] Definición de Planificación de Sprint

prueba Scrum, comparte experiencias y habilidades con el fin de un crecimiento en equipo y mejora del proceso de pruebas que el equipo maneja.

En un nivel 3 se pone en evidencia una organización del equipo de pruebas, se desarrolla un proceso estándar para guiar la ejecución de las pruebas y se maneja una mejora sobre las actividades que se practican. Se entiende la importancia que hay tras las revisiones y el control de calidad por lo cual un programa de preparación del personal se instaura para obtener una mejor preparación del personal del equipo.

La madurez alcanzada durante el nivel 2, se refleja en el nivel 3, en la creación de un plan maestro de pruebas empleando mejores técnicas, desarrolladas por los profesionales en pruebas, formados por la implantación del proceso en el equipo de pruebas. La mejora en las habilidades del equipo y profesionalización de sus actividades en el proceso, permite que el equipo expanda sus técnicas de prueba hacia la evaluación de requerimientos no funcionales.

3.2.2.1. REQUISITOS A CUMPLIR

- Lograr una organización del equipo bajo la estandarización de un proceso de pruebas base.
- Institucionalizar el proceso estándar de pruebas
- Trabajo colaborativo de los miembros del equipo para ejecutar pruebas conjuntas.
- Integrar el proceso estándar de pruebas a lo largo del ciclo de vida del desarrollo del software.
- Capacitar al personal para ampliar y mejorar sus técnicas de pruebas.
- Compartir conocimiento y experiencia con el equipo para la mejora y crecimiento colectivo.
- Definir compromisos e impulsar una visión a la mejora personal y colectiva en el equipo de pruebas.
- Identificar las funcionalidades de las pruebas, descripción del trabajo de pruebas y establecer las funciones que se deben cubrir por el equipo.
- Evaluar la organización del proceso de pruebas dentro del equipo.
- Identificar los puntos de mejora y planificar la implementación de las mejoras que se puedan aplicar al proceso.

- Ejecutar el proceso estándar de pruebas y realizar una evaluación y monitoreo del cumplimiento del mismo.

3.2.2.2. TÉRMINOS A MANEJAR

- **Proceso Estándar de pruebas:** Un proceso estándar corresponde a una serie de actividades establecidas con el fin de optimizar los recursos empleados para la ejecución de las mismas y de esta manera garantizar que los resultados del proceso sean repetibles en futuros proyectos.

3.2.2.3. CRITERIOS A MANEJAR

- **Definir un proceso estándar de pruebas:** El definir los estándares para un proceso de pruebas infiere el establecer las actividades que se deben desarrollar, las posibles técnicas a aplicar y los parámetros que las pruebas deben manejar. Además, se establece que resultados se deben esperar de acuerdo al objeto de prueba y que entregables se deben producir como resultado de la ejecución del proceso.

  En base a la experiencia y madurez alcanzada, se genera un esquema del proceso que cubre los criterios establecidos tanto en el nivel 2 como los necesarios para cumplir el nivel 3 de madurez. En este nivel de madurez se conoce que actividades, recursos y prácticas aportan de mejor manera al proceso permiten alcanzar una mejora personal y del proceso.

- **Definir un programa de capacitación y mejora del personal:** La especialización del conocimiento y la mejora de las habilidades y capacidades técnicas del personal impulsan a que el grupo se consolide y fortalezca, garantizando que las actividades de pruebas realizadas tendrán el respaldo no solo de experiencias individuales sino de un conocimiento impartido y de capacitaciones que potencialicen su trabajo.

  Con el objetivo de impulsar al equipo, la preparación del personal es una inversión para el crecimiento de la organización. Es en este punto donde cursos, charlas y

conversatorios, entre otros, se vuelven aporte y herramientas para alcanzar una mayor madurez del proceso.

- **Definir las funciones, roles y asignaciones que el equipo de pruebas deba cubrir:** La metodología de desarrollo Scrum plantea que un equipo de trabajo dentro de la misma es multidisciplinario, con ello es imperativo que las funciones, roles y asignaciones de cada rol, sean compartidas con todo el equipo de pruebas; el equipo debe conocer los detalles de cada función, rol y asignación para poder aportar cumpliéndolas durante el proceso indistintamente de una delegación directa.

  Las funciones corresponden a las actividades que se deben realizar, los roles son los cargos dentro del equipo de pruebas y las asignaciones son el enlace entre las funciones y los roles, ya que las asignaciones son las que determinaran que rol es el responsable de ejecutar una o varias determinadas funciones y generar uno o varios artefactos resultados de las pruebas aplicadas.

- **Profesionalización del proceso de pruebas:** Las actividades relacionadas a un proceso de pruebas involucran muchos factores como son la experiencia, el ingenio y el conocimiento. Para alcanzar una especialización y mejora de las habilidades de los miembros del equipo de prueba significa que este debe prepararse y mejorar sus capacidades, por eso se lleva a buscar la profesionalización de las actividades de pruebas y no considerarlas como tareas secundarias sino como actividades complementarias al proceso.

3.2.2.4. ARTEFACTOS A GENERAR

- Plan Maestro de Pruebas
- Esquema del Diseño de Pruebas
- Esquema de Casos de Prueba

3.2.3. TMMi Nivel 4 – Medido:

La madurez desarrollada en los niveles 2 y 3 han ayudado a definir una infraestructura técnica administrable, capaz de ejecutar pruebas complejas y soportar una mejora del proceso de pruebas. En el nivel de madurez 4, el proceso de pruebas es medible y alienta a alcanzar un crecimiento futuro del mismo, ya que es un proceso bien definido e integra un programa de mediciones que permite evaluar la calidad del proceso de pruebas, la productividad y monitorear las mejoras.

Las mediciones en este nivel, sirven para la toma de decisiones en base a los resultados, con esto implementar un proceso de evaluación de calidad del producto definiendo las necesidades y atributos de la calidad.

3.2.3.1. REQUISITOS A CUMPLIR

- Definir proceso de medición del proceso de pruebas.
- Definir la medición del proceso como evidencia del cumplimiento del mismo.
- Revisiones avanzadas para trasformar la detección de defectos en prevención de defectos.
- Recolectar los resultados de las pruebas ejecutadas para conformar un historial de resultados.
- Aprender de las ejecuciones para mejorar los procesos y procedimientos en las pruebas.

3.2.3.2. TÉRMINOS A MANEJAR

- **Medidas para Pruebas**: Corresponde a los parámetros que se pueden establecer como evidencia cuantitativa y cualificativa del proceso o la ejecución de pruebas. A estas medidas se las entiende como los datos obtenidos de las pruebas que permiten marcar los criterios de comparación de resultados, que determinan la aprobación o no de los principios de medida y así plantear acciones correctivas, preventivas o de mejora.

- **Principios de Medida:** Criterios que toman como base a las medidas de prueba para considerar y establecer los factores evaluables resultantes de la ejecución de una prueba. Estos principios establecen bajo que pautas una prueba se la mide para que los resultados obtenidos de la misma aporten un valor al proceso.

- **Error, Defecto, Fallo:** El *error* es la acción o actividad humana que causa un comportamiento o resultado incorrecto del sistema, el *defecto* va a ser el resultado de la incorrecta acción humana y esto como consecuencia produce un *fallo*, el cual es la manifestación física o funcional del defecto causado por el error humano.

- **Prevención de defectos:** Acciones y medidas tomadas con el fin de que los defectos conocidos o comunes sean repetidos. Al igual que permitan determinar patrones que posibiliten mitigar futuros errores.

3.2.3.3. CRITERIOS A MANEJAR

- **Definir principios para medir las pruebas:** Establecer que factores representan los resultados de las pruebas, para en base a esos resultados construir los principios de evaluación. Las medidas de las pruebas comprenden a los valores cuantitativos o características cualificativos que determinan los puntos de consideración dentro de los criterios. Los principios serán las conclusiones de los resultados de las pruebas que permitan fundamentar sobre las mismas acciones a tomar, sean estas con el fin de realizar correcciones al producto o mejoras al proceso de evaluación ejecutado.

- **Establecer entregables como resultado a las pruebas:** Los documentos entregables que contengan el detalle de las pruebas ejecutadas y los resultados de las mismas componen un histórico para retro alimentación de las pruebas y como base para futuras ejecuciones. La evaluación y tabulación de los resultados de las pruebas permiten establecer conclusiones sobre la ejecución de las mismas y esto a su vez documentado adecuadamente funciona como fuente de asesoría para la toma de decisiones sobre problemas afines, corrección temprana de problemas similares y prevención en etapas tempranas del proceso.

En este punto del proceso de pruebas, los entregables de las mismas facilitan la evaluación de requerimientos similares, al igual que la re-ejecución de casos de pruebas. Basándose en un histórico de pruebas ejecutadas y resultados obtenidos permite la optimización y mejor manejo del tiempo de prueba al disponerse de una base de conocimiento, y con ello asegurar la repetición de un nivel de calidad en las ejecuciones.

- **Establecer los criterios para definir error, defecto y fallo:** Los criterios para definir las diferencias entre error, defecto y fallo, permite a su vez identificar los causales y por qué se suscitan los mismos. Puntualizar claramente las diferencias entre error, defecto y fallo aporta a determinar los puntos de mejora para un proceso de pruebas más nítido y eficiente, ya que se pueden aplicar pruebas que abarquen de mejor manera los requerimientos y evalúen más eficientemente el cumplimiento o no de los mismos.

- **Definir criterios para prevenir defectos:** El resultado principal de la ejecución de una prueba es detectar defectos en la funcionalidad del objeto de prueba o el no cumplimiento de un requerimiento, el siguiente paso en la mejora es la prevención de defectos, paso en donde tomando en cuenta la experiencia y conocimiento del entorno y la identificación de patrones de defectos, se puede llegar a mitigar la aparición de defectos conocidos.

  Esto se llega identificando, preferentemente a tempranas etapas, la raíz de los fallos al igual que los causales de los defectos encontrados durante el proceso de pruebas. La prevención de defectos actúa sobre el conocimiento y experiencia adquirida de la repetición efectiva de un proceso de pruebas completo y documentado que respalda la reproducción eficiente del proceso.

3.2.3.4. ARTEFACTOS A GENERAR

- Plan Maestro de Pruebas
- Esquema del Diseño de Pruebas
- Esquema de Casos de Prueba

- Lista de Verificación

3.2.4. TMMi Nivel 5 – Optimizado:

En un nivel 5 de madurez, la organización del equipo de pruebas es capaz de continuamente mejorar su proceso en base al conocimiento, experiencia y adquisición de nuevas técnicas, Esto partiendo de la capacitación y mejora de las habilidades del personal.

La optimización del proceso se basa en un proceso estandarizado, manejable, bien definido y efectivo, respaldado por la documentación resultado de la aplicación eficaz y eficiente de las técnicas de pruebas mejoradas.

3.2.4.1. REQUISITOS A CUMPLIR

- Enfoque en la prevención de defectos más en la detección de los mismos.
- Soportar el proceso con herramientas de automatización tanto como los recursos y el entorno lo permita.
- Soportar el reúso de material para las pruebas.
- Enfoque a lograr una mejora continua.

3.2.4.2. TÉRMINOS A MANEJAR

- **Automatización**: Se entiende como la utilización de herramientas que permitan que un proceso repetitivo sea ejecutado de manera autónoma con el fin de optimizar recursos tales como el tiempo y el personal a cargo de la ejecución.
  La automatización de proceso repetitivo aporta en ejecutar dicho proceso con una mínima intervención y generando un informe de resultados que sea analizable y medible. Las herramientas de automatización son desarrolladas y ajustadas de acuerdo a las necesidades del proceso de pruebas.

- **Entorno de Prueba**: Es el ambiente controlado donde la ejecución de las pruebas se puede realizar con un factor de confianza sobre las condiciones del entorno y los resultados obtenidos. Un entorno de prueba busca emular un ambiente de producción

donde el objeto de prueba se nutre de datos semejantes a los reales que posibilitan que los resultados obtenidos sean una representación de los posibles resultados a obtener de un ambiente donde actué el cliente final.

- **Mejora continua**: Principio por el cual se busca que toda actividad y proceso presente resultados que puedan ser analizados y medidos con el fin de determinar puntos donde el proceso o las actividades se puedan mejorar. Se determina como mejora continua en base a que la búsqueda de puntos de mejora es persistente y repetida con el fin de llegar a una optimización de recursos y procesos.

3.2.4.3. CRITERIOS A MANEJAR

- **Determinar procesos automatizables**: Identificar los procesos repetitivos dentro de las actividades de pruebas permite determinar si estos son automatizables o no. El automatizar procesos repetitivos conlleva evaluar si el esfuerzo e inversión, tanto de recursos como de tiempo, representa una inversión apropiada frente a los resultados y optimizaciones a obtener.

  El desarrollar herramientas de automatización implica que el personal a cargo tiene que poseer o cultivar habilidades que vayan en pro de las actividades de pruebas en sí, al igual debe destinarse los recursos necesarios para un desarrollo eficiente de dichas herramientas y sobre manera manejar un conocimiento específico de los detalles del proceso a automatizar.

- **Definir condiciones y criterios para el reuso**: El reuso de material de pruebas es una herramienta muy útil al momento de optimizar recursos y tiempo, para manejar un adecuado proceso de reúso se debe considerar detalles previos y post a la reutilización del material de prueba.

  El fin del reuso es la optimización de recursos, por ello se debe evaluar los motivos y resultados que se llegaría a obtener si se reutiliza material de otras pruebas, si tras el análisis de costo-beneficio se determina que el reuso aportará valor y optimizará

recursos, se debe establecer las condiciones en que se efectuara el reuso, cerciorarse de que los cambios y reutilización de recursos sea documentado para dejar evidencia de las acciones tomadas, el porqué de las mismas y los resultados obtenidos de dichas acciones tomadas.

- **Criterios para determinar un proceso de mejora continua**: La mejora continua es una herramienta muy útil al momento de buscar el crecimiento y superación del equipo de pruebas. Por ello se debe determinar qué acciones, métodos y controles se deben efectuar con el fin de que los resultados del proceso sirvan de semillas para ejecutar acciones de mejoras y perfeccionamiento en la repetición del proceso de pruebas.

3.2.4.4. ARTEFACTOS A GENERAR

- Plan Maestro de Pruebas
- Esquema del Diseño de Pruebas
- Esquema de Casos de Prueba
- Lista de Verificación
- Catálogo de Pruebas

3.3. Definición de lineamientos y guías del Marco de Referencia de Calidad.

3.3.1. Detalle de Artefactos a Generar

Las siguientes descripciones corresponden a los artefactos necesarios para el proceso de pruebas, en dichos artefactos se evidencia todo el avance y desarrollo del proceso y además se recopila datos de los resultados de las pruebas ejecutadas y de los procesos desarrollados que permiten tener una base del conocimiento aplicable para futuros proyectos.

3.3.1.1. Plan Maestro de Pruebas

El propósito del Plan Maestro de Pruebas es proveer un plan de pruebas y una administración de la documentación de las mismas, frente a todo el desarrollo del proyecto, hasta diferentes proyectos que sean componentes de un proyecto más complejo. El plan maestro de pruebas permite además tener un enfoque claro de los requerimientos del software y la planificación del aseguramiento de la calidad que pretenda que el proceso aporte al producto final. Además, el plan maestro de pruebas permite una clara identificación de los componentes de un sistema, los requerimientos de los mismos y la distribución de los recursos tanto de tiempo como de personal e infraestructura. En este documento de pruebas igual se definen los objetivos a alcanzar con la finalización efectiva del mismo.

Componentes del Plan Maestro de Pruebas. (Ver Plantilla)

A continuación, se describen las secciones del Plan Maestro de Pruebas que corresponden a las partes críticas del plan, las cuales deben ser descritas de la mejor manera para aportar al conocimiento colectivo y mejora continua.

1. *Control de cambios:* Tomando la información de la base del conocimiento que conforma el Catálogo de Pruebas se define cuáles fueron los últimos cambios que el proyecto tuvo y cuál fue el impacto de dichos cambios.
2. *Datos Informativos del Proyecto*: Detalles y características del proyecto sobre el cual se va a ejecutar el proceso de pruebas y detalles del proceso de pruebas en sí que se va a ejecutar.
    - *2.1.Requerimientos Funcionales:* Inferidos de los requerimientos estipulados para el proyecto en su inicio, frente al cumplimiento de estos requerimientos se busca la ejecución adecuada del proceso de pruebas para asegurar el correcto cumplimiento de los mismos.
    - *2.2.Requerimientos No Funcionales*: Requerimientos a evaluar del proyecto relacionados a su desempeño, velocidad y funcionamiento fuera de los requerimientos estipulados en un inicio.

2.3. *Antecedentes*: Datos históricos del proyecto que sean de relevancia para el proceso de pruebas. Se detalla la ubicación de documentación externa que pueda ser relevante para el proceso.

2.4. *Dependencias*: Información sobre objetos prueba o módulos del sistema que tengan relación con el proyecto y que puedan ser un factor a considerar al ejecutar el proceso de pruebas.

2.5. *Cronograma*: Calendario con la distribución del tiempo, recursos y responsables de cada etapa y tipos de pruebas que se vayan a ejecutar durante el proceso de pruebas.

2.6. *Consideraciones para la prueba*: En esta sección del plan maestro de prueba se debe detallar los objetos de pruebas que se van a evaluar durante el proceso de pruebas, al igual que detalles como tipos de objetos de prueba, prioridades, recursos tanto de hardware, software, personal e infraestructura que sean necesarios para ejecutar un adecuado proceso. Y además definir los riesgos que se puedan enfrentar durante al proceso de pruebas,

3. *Información del Plan Maestro de Prueba*: En esta sección del Plan Maestro de Pruebas, se debe documentar, con tanto detalle cómo se considere necesario, las actividades que se van a ejecutar durante el proceso de pruebas, al igual que el o los responsables de las mismas y de los artefactos a generar.

   Se debe analizar y documentar los requerimientos que pretendan ser evaluados durante el proceso de pruebas de manera clara y concisa, a modo de poder traducirlos en términos de acciones. Con esto en base a los requerimientos evaluados se diseñan las pruebas que cubran dichos requerimientos con el fin de evaluar eficientemente el cumplimiento de los mismos.

   3.1. *Proceso: Operación:* En base al análisis de los requerimientos no funcionales del sistema se ejecutan pruebas complementarias al proceso de prueba, que permita determinar de mejor manera el funcionamiento adecuado del mismo.

   3.2. *Proceso: Mantenimiento:* Una vez completado el proceso de desarrollo y el objeto de prueba o proyecto fueron puestos en producción, son necesarias pruebas de mantenimiento para corroborar y garantizar que el sistema se

mantenga en un correcto funcionamiento, del sistema y su respuesta a las demandas del entorno operativo.

4. *Requisitos de documentación para la prueba:* Detalle de que documentación es requerida como evidencia para la eficiente ejecución del proceso de pruebas.
5. *Requisitos administrativos para la prueba:* Detalle de los requerimientos que la administración requiera que se cumplan o consideren. Estos pueden estar ligados a requerimientos del negocio más no de funcionalidad.
6. *Requisitos de reporte de resultados de prueba:* Detalle de los artefactos e información que se requiere generar a lo largo de proceso de pruebas
7. *Informe de Resultados del Plan Maestro de Prueba*: En la sección de informe de resultados del Plan Maestro de Pruebas se debe documentar los resultados de las pruebas ejecutadas a lo largo de todo el proceso de pruebas. El nivel de detalle debe ser consistente a la complejidad e importancia que el proyecto tenga ya que está información aporta al desarrollo del Catálogo de Pruebas.

3.3.1.2. Esquema del Diseño de Pruebas

El Esquema del Diseño de Pruebas tiene como propósito especificar el enfoque de la prueba frente a cualquier requerimiento a evaluar. En este esquema se identifica los objetos de prueba a ser probados y la asociación que los mismos tienes entre sí y con otras dependencias.

Componentes del Diseño de Pruebas (Ver Plantilla)

1. *Control de cambios:* Tomando la información de la base del conocimiento que conforma el Catálogo de Pruebas se define cuáles fueron los últimos cambios que el proyecto tuvo y cuál fue el impacto de dichos cambios.
2. *Introducción*: Se identifican los objetos de pruebas que van a ser evaluados, de igual manera se define la información de alcance y referencias que el objeto de prueba pueda tener, con el fin de que, cada uno sea claro y comprensible su funcionamiento dentro del sistema

Detalle del Diseño de Pruebas: En la sección de detalle de diseño se debe especificar las pruebas que se van a ejecutar, para ello se debe definir cada uno de los objetos de prueba, los criterios de entrada que cada objeto maneja y los criterios de salida esperados como resultado de la ejecución de la prueba. Como evidencia del proceso ejecutado se debe establecer los entregables que sean necesarios de acuerdo al nivel de profundidad de las pruebas.

3.3.1.3. Esquema de Casos de Prueba

El propósito del Esquema de Casos de Prueba, es definir, en un determinado nivel de detalle, la información necesaria para la ejecutar la evaluación de un determinado objeto de prueba o grupos de objetos y sus asociaciones entre sí. Para este esquema se debe definir los datos de entrada y los resultados de salida esperados de la prueba bajo una o varias condiciones a evaluar. Se recomienda generar un documento de Esquema de Casos de Prueba por cada objeto de prueba ya que de este modo se puede llevar un mejor control y detalle de la información correspondiente a cada objeto.

Componentes del Esquema de Casos de Prueba (Ver Plantilla)

1. *Control de cambios:* Tomando la información de la base del conocimiento que conforma el Catálogo de Pruebas se define cuáles fueron los últimos cambios que el proyecto tuvo y cuál fue el impacto de dichos cambios.
2. *Introducción*: Como parte de la introducción del esquema de caso de prueba se debe describir en mayor detalle cada caso de prueba que se vaya evaluar, de igual manera información como el alcance y la prioridad de dicho objeto es relevante para tener un concepto claro del objeto.
3. *Detalles*: La sección de detalle del esquema de caso de prueba es muy relevante, ya que es en esta sección que se van a definir las características que la prueba debe cumplir para dar por completado el objeto de prueba. Igualmente se debe definir los datos de entrada y de salida que debe manejar el objeto de prueba, esto enmarcado a las técnicas de prueba que se vaya a aplicar al objeto.

Unos factores que se deben considerar y especificar en esta sección, son los requerimientos del entorno y procedimientos especiales que el objeto de prueba necesita para evaluar su correcto funcionamiento, igualmente definir las interdependencias que un objeto pueda tener con otros objetos o con algún agente externo.

4. *Flujo Normal*: Corresponde al camino ideal que se espera que el objeto de prueba siga. En base a este camino ideal esperado se evalúa el cumplimiento del requerimiento. Dentro del flujo normal se definen las acciones que se deben ejecutar y los resultados que se esperan como respuesta. Alrededor del flujo normal se plantean las pruebas complementarias que ayudan a corroborar el correcto funcionamiento del objeto de prueba.
5. *Flujo Alterno*: El flujo alterno corresponde a las acciones que se pueden aplicar al objeto de prueba para esperar resultados alternos al flujo normal, Las acciones que se ejecutan están contempladas fuera de la definición de los requerimientos.

3.3.1.4. Lista de Verificación

La Lista de Verificación, tiene como objeto el llevar una bitácora del cumplimiento o no de los requerimientos de la prueba en ejecución, esta lista aporta una herramienta importante al momento de generar un reporte de una incidencia. La lista de verificación permite llevar control de qué puntos del objeto de prueba, durante la evaluación, fueron aprobados sin ninguna novedad, al igual que cuales puntos del objeto de prueba no fueron aprobados y necesidad ser corregidos.

Esta lista registra, de igual manera los casos de prueba relacionados a la misma, Este artefacto junto con el Esquema de Casos de Prueba, conforman de primera mano la evidencia del proceso de prueba ejecutado y la información recolectada durante la prueba.

Componentes de la Lista de Verificación (Ver Plantilla)

1. *Introducción*: Como parte de la introducción de la Lista de Verificación se debe describir en detalle las validaciones y criterios que se deben evaluar, de igual manera

información particular del objeto de prueba, la lista de verificación permite llevar un control de las tareas de pruebas ejecutadas y su duración para posteriormente realizar evaluaciones del proceso en pruebas en sí. En esta sección se debe registrar el nombre del probador que ha diseñado la lista en un inicio.

2. *Verificaciones*: Las verificaciones corresponden a una lista de validaciones que se deben evaluar al momento de probar un objeto, estas validaciones manejan dos criterios de acuerdo a si se cumplió o no, junto a algún comentario que se tenga que documentar frente a la validación. Al final de la lista de verificaciones se debe registrar quién fue el probador encargado que evaluó los criterios de la lista.
3. *Detalle de la prueba*: La sección de detalle describe los resultados de la prueba relacionados con el cumplimiento o no de las validaciones, igual se registran los comentarios que el probador encargado pueda tener frente a la prueba ejecutada.

3.3.1.5. Catálogo de Pruebas

El objetivo del Catálogo de Pruebas es llevar de manera organizada un historial de los procesos de evaluación ejecutados sobre distintos objetos de prueba, con el fin de que esta información recopilada sirva de respaldo para futuros procesos de prueba y como base de conocimiento sobre métodos de pruebas aplicables en futuros proyectos

Componentes del Catálogo de Pruebas (Ver Plantilla)

1. *Índice de Pruebas*: Este artefacto permite administrar de manera ordenada un listado de los objetos de pruebas pertenecientes a distintos proyectos que hayan sido evaluados bajo el proceso de pruebas. Igualmente, este artefacto registra las fechas en las que dichos procesos se ejecutaron a manera de llevar un historial de las pruebas y de los proyectos trabajados, por otro lado, se registra la ubicación ya que, con la idea de que esta información sea accesible para cada miembro del equipo, la misma debe ser organizada sea bien de manera digital o física en un determinado lugar

De ser de manera digital se recomienda el definir un servidor, propio o externo, donde cada miembro del equipo tenga acceso a ver mas no editar la información recopilada en el Catálogo de Prueba, para editar es mejor descargar una copia, modificarla y subir al repositorio una nueva versión parra así manejar criterios de versionamiento.

3.3.1.6. Reporte de Incidencias.

Una incidencia dentro del presente marco de referencia, se la define como todo resultado erróneo de una prueba frente al requerimiento que se esté evaluando, es decir, una incidencia puede ser cualquier error, defecto o fallo que un objeto de prueba pueda presentar que impida que el mismo cumpla su correcto funcionamiento de la manera que se espera. También una incidencia puede ser algún factor externo o ajeno al objeto de prueba que de alguna forma influya en el resultado de la prueba o impida el cumplimiento del proceso.

Con esto se establece que el reporte de incidencia, correspondería a un artefacto en el cual se detalle cual fue la incidencia encontrada durante el proceso de prueba, las consecuencias que la incidencia pudo causar y aún más importante las posibles causas que provocaron la incidencia, esto junto a un detalle de como evaluar la incidencia y las acciones correctivas necesarias conforman el Reporte de Incidencias.

Componentes del Reporte de Incidencias. (Ver Plantilla)

1. *Control de cambios:* Tomando la información de la base del conocimiento que conforma el Catálogo de Pruebas se define cuáles fueron los últimos cambios que el proyecto tuvo y cuál fue el impacto de dichos cambios.
2. *Introducción*: La sección de introducción del Reporte de Incidencias, registra toda la información relevante del hallazgo, como las características de la incidencia encontrada y las condiciones en las que la misma fue descubierta.
3. *Detalles*: La sección de detalle del reporte de incidencia documenta todo el impacto que la incidencia tuvo en el entorno. Es en esta sección donde, con el mayor detalle posible, se debe registrar la mayor cantidad de información, que pueda ayudar a

determinar por qué se suscitó y posteriormente inferir las medidas necesarias para su corrección

3.3.1.7. Reporte de Objeto Prueba

El Reporte de Objeto de Prueba es un artefacto que contendrá el detalle de un objeto de prueba que fue sometido a un proceso de evaluación, Este artefacto se lo completa en base a la información generada tras evaluar un caso de prueba sobre un determinado objeto que cumpla con la definición de objeto de prueba.

Componentes del Reporte de Pruebas. (Ver Plantilla)

1. *Control de cambios:* Tomando la información de la base del conocimiento que conforma el Catálogo de Pruebas se define cuáles fueron los últimos cambios que el proyecto tuvo y cuál fue el impacto de dichos cambios.
2. *Introducción*: La sección de introducción del Reporte de Prueba, registra toda la información relevante sobre la prueba o pruebas realizadas, el detalle de la información debe ser alto, ya que, de esta manera, esta información aporta al Catálogo de Pruebas para consolidar una base de conocimiento aplicable y repetible en proyectos futuros.
3. *Detalles*: La sección de detalle del reporte de objeto de prueba registra todos los pasos realizados y resultados obtenidos como parte de la evaluación del objeto de prueba. Es en esta sección donde, en el mayor detalle posible, se debe registrar la mayor cantidad de información, que aporte para generar conceptos y técnicas de prueba aplicables en futuros proyectos.

3.3.2. TMMi Nivel 2 – Manejado: Desarrollo del Proceso

LINEAMIENTOS PARA CUMPLIR LOS REQUISITOS

*Nivel 2 Lineamiento #

N2L1*: Catalogar el tipo de objeto de prueba.

1. Identificar el tipo de objeto de prueba de acuerdo a la descripción.
2. Definir requisitos necesarios para evaluar el objeto de prueba.
3. Definir orden para probar los distintos objetos de prueba.
4. Separar por objetos de prueba que se relacione de acuerdo a funcionalidad cumplan.

N2L2: Procedimientos para probar de acuerdo al tipo de objeto de prueba.

1. **Objeto de Prueba Nuevo**: Revisar los requerimientos a evaluar, las validaciones que tenga que cumplir y preparar los requerimientos de entorno que necesite para ser probado.

1.1. Objeto de Capa de Base de Datos.

1.1.1. Evaluar nombres de tablas a usar de acuerdo a especificación.
1.1.2. Evaluar nombre de columnas de acuerdo a especificación.
1.1.3. Evaluar tipos de datos de acuerdo a especificación.
1.1.4. Evaluar claves primarias y secundarias de acuerdo a especificación.
1.1.5. Evaluar relación entre tablas de acuerdo a especificación.
1.1.6. Ejecutar desde la base, sentencias de inserción, consulta, modificación y eliminación de acuerdo a especificación,
1.1.7. Ejecutar desde la base, sentencias que prueben las relaciones entre tablas de acuerdo a especificación.

1.2. Objeto de Capa de Usuario.

1.2.1. Evaluar cumplimiento de requerimientos estéticos.

1.2.2. Evaluar procedimientos de inserción, consulta, modificación y eliminación de acuerdo a especificación,

1.2.3. Evaluar validaciones de campos.

1.2.4. Evaluar tipos de datos por los campos.

1.2.5. Evaluar condiciones de inserción, consulta, modificación y eliminación de acuerdo a especificaciones del negocio.

2. **Objeto de Prueba Antiguo:** Revisar los cambios en los requerimientos, las validaciones nuevas o cambios que tenga que cumplir y preparar los requerimientos de entorno que necesite para ser probado.

2.1. Objeto de Capa de Base de Datos.

2.1.1. Evaluar cambios de nombres de tablas a usar de acuerdo a especificación.

2.1.2. Evaluar cambios de nombre de columnas de acuerdo a especificación.

2.1.3. Evaluar cambios de tipos de datos de acuerdo a especificación.

2.1.4. Evaluar cambios de claves primarias y secundarias de acuerdo a especificación.

2.1.5. Evaluar cambios de relación entre tablas de acuerdo a especificación.

2.1.6. Ejecutar desde la base, sentencias de inserción, consulta, modificación y eliminación de acuerdo a especificación,

2.1.7. Ejecutar desde la base, sentencias que prueben los cambios en las relaciones entre tablas de acuerdo a especificación.

2.2. Objeto de Capa de Usuario.

2.2.1. Evaluar cumplimiento de los cambios de requerimientos estéticos.

2.2.2. Evaluar cambios en procedimientos de inserción, consulta, modificación y eliminación de acuerdo a especificación,

2.2.3. Evaluar cambios en validaciones de campos.

2.2.4. Evaluar cambios de tipos de datos por los campos.

2.2.5. Evaluar cambios en las condiciones de inserción, consulta, modificación y eliminación de acuerdo a especificaciones del negocio.

N2L3: Diseño de los casos de prueba en base a requerimientos funcionales del producto del software.

1. Identificar los requerimientos por objeto de prueba.
2. Definir los datos de entrada necesarios para la prueba.
3. Identificar los pasos para probar la funcionalidad a evaluar.

N2L4: Niveles de profundidad de las pruebas a ejecutar.

1. Definir la criticidad del objeto de prueba.
2. Definir la complejidad de probar el objeto de prueba.
3. Determinar el nivel de profundidad entre:
    a. *Bajo*: Objeto poco crítico y poco complejo de probar.
    b. *Medio*: Objeto medio crítico y poco complejo de probar.
    c. *Alto*: Objeto crítico y complejo de probar.

N2L5: Cronograma de ejecución de las pruebas.

1. Ordenar los objetos de prueba de acuerdo a complejidad y criticidad.
2. Estimar tiempos necesarios para probar cada objeto de prueba ordenado.
3. Calendarizar los objetos de prueba de acuerdo al orden y el tiempo que lleve probar cada objeto.

N2L6: Monitoreo y control de ejecución de pruebas.

1. Crear lista de todos los objetos de prueba que están en cola para ser probados.
2. Revisar el estado del proceso de pruebas para cada objeto de prueba.
3. Registrar el estado de cada objeto de prueba.
4. Controlar la finalización del proceso de pruebas por cada objeto evaluado.

3.3.3. TMMi Nivel 3 – Definido: Desarrollo del Proceso

LINEAMIENTOS PARA CUMPLIR LOS REQUISITOS

***Nivel 3 Lineamiento #**

N3L1*: Organización del equipo bajo la estandarización del proceso de pruebas base.

Para la organización del equipo de trabajo, definición de roles y asignación de tareas, el estándar *ISO/IEC 29110*[44], apoya a la propuesta con las actividades, tareas y roles necesarias para la gestión equipo de trabajo.

1. Definir roles dentro del equipo de pruebas.
2. Socializar técnicas para pruebas.
3. Definir repositorios para la documentación a generar.
4. Definir estrategias de trabajo individuales y colectivas.
5. Definir procedimientos de comunicación en el grupo.

N3L2: Ejecución conjunta de pruebas.

1. Identificar necesidades de ejecutar pruebas conjuntas.
2. Definir objetivos de las pruebas conjuntas.
3. Definir estrategias para optimización de recursos para las pruebas.
4. Coordinar tiempos entre miembros del equipo para pruebas conjuntas.
5. Coordinar el trabajo para que el mismo no sea remetido.
6. Establecer canales de comunicación dentro del Equipo Scrum.

N3L3: Institucionalizar proceso estándar de pruebas.

1. Definir un proceso de pruebas tomando en cuenta el aprendizaje obtenido en el Nivel 2 de Madurez.

[44] Estándar ISO/IEC 29110

2. Socializar el proceso estándar de pruebas desarrollado en base a la madurez adquirida durante el Nivel 2 de Madurez y los lineamientos establecidos en el Nivel 3 de Madurez.
3. Definir los procedimientos para evaluar cada tipo de objeto de prueba considerando lo definido en el Nivel 2 de Madurez.
4. Definir el flujo para el desarrollo del proceso estándar de pruebas.
5. Documentar los lineamientos establecidos para el proceso estándar de pruebas para facilitar su socialización y repetición continua en todos los proyectos.

N3L4: Ejecución el proceso estándar de pruebas

1. Paralelo la planificación de Sprint, realizar la planificación de pruebas para el Sprint[45].
2. Generar el Plan Maestro de pruebas[46].
3. Organizar al personal para la distribución de las tareas de pruebas.
4. Delegar tareas específicas a un probador dentro del equipo.
5. Diseñar las pruebas a ejecutar de acuerdo a los objetos de prueba.
6. Documentar en el artefacto Esquema de Diseño de Pruebas[47].
7. Definir los casos de prueba de acuerdo a las pruebas diseñadas.
8. Documentar en el artefacto Esquema de Casos de Prueba[48].
9. Preparar el entorno de prueba y los requerimientos del sistema necesarios para prueba.
10. Elaborar las Listas de Verificación[49].
11. El probador debe mantener comunicación con el desarrollador encargado para estar al tanto del estado del desarrollo de los objetos de prueba.
12. Una vez que los objetos de prueba pasen de desarrollo a pruebas, el probador debe actualizar el estado de los casos de prueba.

[45] Definición de Sprint Planning
[46] Plantilla de Plan Maestro de Pruebas
[47] Plantilla de Esquema de Diseño de Pruebas
[48] Plantilla de Esquema de Casos de Prueba
[49] Plantilla Listas de Verificación

13. Completar la información faltante en los Esquemas de Casos de Uso y en las Listas de Verificación correspondientes.
14. Documentar los procedimientos en el Esquema de Casos de Uso.
15. Ejecutar el proceso estándar de pruebas definido por el Equipo Scrum[50] de pruebas.
16. Durante el Sprint comunicar diariamente, el progreso, el estado y los impedimentos de las pruebas en las reuniones de coordinación[51].
17. Completar la evaluación de los objetos de prueba.
18. Reportar Incidencias encontradas durante el proceso de pruebas empleando el artefacto de Reporte de Incidencias[52].
19. Reportar los resultados de las pruebas de manera oral y empleando el artefacto de Reporte de Objeto de Prueba[53].
20. Al terminar el proceso de pruebas, actualizar el Plan Maestro de Pruebas con la información generada.

N3L5: Capacitación al personal y socialización de conocimiento y experiencias.

1. Impartir conversatorios con personal capacitado para mejorar habilidades.
2. Reuniones del equipo de pruebas para compartir experiencias.
3. Compartir lecturas relacionadas a metodologías, técnicas o herramientas para pruebas que puedan ser de interés común para el equipo.
4. Desarrollar un programa de entrenamientos que aporta a mejorar las habilidades individuales y colectivas del equipo de pruebas.
5. Incentivar al dominio del conocimiento sobre el producto que la organización desarrolle o la línea de trabajo que la organización maneje.

N3L6: Compromisos personales y colectivos.

[50] Definición de Scrum Team
[51] Definición de Stand Up Meeting
[52] Plantilla de Reporte de Incidencias
[53] Plantilla de Reporte de Objeto de Prueba

1. Compromiso de cada miembro en aportar de la mejor manera para el desarrollo óptimo del proceso de pruebas.
2. Compromisos colectivos de aportar para la mejora colectiva del equipo.

N3L8: Funciones a cubrir por el equipo.

1. Identificar las áreas de trabajo dentro del equipo.
2. Definir líderes de área de acuerdo a las habilidades.
3. Definir roles dentro del equipo de pruebas.
4. Definir perfiles de trabajo para los distintos roles.
5. Definir actividades correspondientes a los roles.

Para el cumplimiento de los siguientes lineamientos, los detalles relacionados a los mismos se los desarrolla en el apartado 4.4. de la presente propuesta.

N3L9: Evaluación de la organización del proceso de pruebas dentro del equipo.

1. Identificar si el proceso estándar de pruebas es conocido por cada miembro del equipo.
2. Solicitar retroalimentación del equipo de pruebas sobre su experiencia trabajando bajo el Proceso Estándar de Pruebas[54].
3. Formar el Catálogo de Pruebas[55] en base a la información recolectada de los proyectos trabajados bajo el Proceso Estándar de Pruebas.
4. Tomando la información recolectada en el Catálogo de Pruebas y la retroalimentación por parte del equipo de prueba, estableces parámetros que identifiquen el nivel de organización del proceso de pruebas dentro de las actividades del equipo.

[54] Definición de Retrospective.
[55] Definición de Catálogo de Pruebas

N3L10: Puntos de mejora del proceso.

1. En base a la retroalimentación del equipo de pruebas, identificar puntos de mejora del proceso de pruebas.
2. Convertir los comentarios en objetivos a mejorar dentro del proceso.
3. Identificar fortalezas y debilidades del proceso.
4. Diseñar acciones que permitan optimizar los puntos de mejora del proceso.

N3L11: Implementación de mejoras al proceso.

1. Analizar la viabilidad e importancia de las acciones correctivas diseñadas.
2. Asignar actividades y responsabilidades para aplicar las mejoras individuales al proceso.
3. Definir procedimientos para la implementación de las mejoras al proceso de pruebas.
4. Aplicar gradualmente las mejoras para que dichas mejoras sean adoptadas por todo el equipo de pruebas.

N3L12: Evaluación y monitoreo del cumplimiento del proceso.

1. Documentar los objetivos que se plantean como mejoras del proceso de pruebas.
2. Registrar el progreso de las actividades destinadas a mejorar el proceso.
3. Monitorear si las acciones planificadas están aportando a la mejora del proceso.
4. Analizar paulatinamente los resultados de las acciones aplicadas para mejorar el proceso estándar de pruebas.
5. Una vez aplicadas todas las acciones planificadas, evaluar el desarrollo de todo el Proceso Estándar de Pruebas.
6. Aplicar el proceso en nuevos proyectos o iteraciones de un mismo proyecto para determinar la mejora o no del proceso de pruebas.

3.3.4. TMMi Nivel 4 – Medido: Desarrollo del Proceso

LINEAMIENTOS PARA CUMPLIR LOS REQUISITOS

Mayor detalle sobre la medición del proceso de pruebas, se la desarrolla en el apartado 4.4 Planteamiento de un Proceso de Evaluación para el cumplimiento del Marco de Referencia de Calidad[56] de la presente propuesta.

*Nivel 4 Lineamiento #

N4L1*: Medición del proceso de pruebas.

1. Plantear criterios de medición para evaluar el progreso del proceso de pruebas.
2. Registrar los resultados de los criterios de medición.
3. Analizar los resultados de los criterios de medición.
4. En base al análisis de resultados de los criterios de medición determinar el progreso del proceso de pruebas.

N4L2: Revisiones avanzadas para la prevención de defectos.

1. Diseñar pruebas conjuntas de mayor nivel para cubrir los requerimientos funcionales del objeto de prueba.
2. Diseñar pruebas conjuntas de mayor nivel para cubrir los requerimientos no funcionales del objeto de prueba. Características de calidad planteadas por el estándar *ISO/IEC 25010*[57].
3. Aplicar técnicas de pruebas especializadas.
4. Ejecutar los tipos de revisiones planteadas por el estándar *IEEE/EIA 1028* [58, 59]

N4L3: Consolidación de resultados de las pruebas ejecutadas.

[56] Planteamiento de un Proceso de Evaluación para el cumplimiento del Marco de Referencia de Calidad
[57] Estándar *ISO/IEC 25010*
[58] Estándar IEEE/EIA 1028
[59] (Software & Systems Engineering Standards Committee, 2008, pp. 6 - 37)

1. Establecer el Catálogo de Pruebas como un respaldo histórico del desarrollo del proceso de pruebas.
2. Tomar la recopilación de resultados de pruebas para determinar errores frecuentes o comunes.
3. Listar errores frecuentes o comunes con sus posibles causas y las acciones correctivas que los mismos admitían.

N4L4: Mejora de procesos y procedimientos en base al aprendizaje.

1. En base a la madurez adquirida en los Niveles 2 y 3 de Madurez, definir un proceso de pruebas mejorado y optimizado.
2. Optimizar el uso de los recursos como: los artefactos de las pruebas, catálogo de pruebas y habilidades adquiridas por los miembros del grupo.
3. En base a la experiencia y desarrollo de habilidades, buscar mejorar las técnicas y procedimientos aplicados a las pruebas.

3.3.5. TMMi Nivel 5 – Optimizado: Desarrollo del Proceso

LINEAMIENTOS PARA CUMPLIR LOS REQUISITOS

*Nivel 5 Lineamiento #,

N5L1*: Prevención de defectos.

1. Buscar la prevención de defectos más no la detección.
2. Tomar la lista de errores conocidos comunes junto con las causas y acciones correctivas y plantear acciones para mitigar estos errores.
3. Analizar riesgos futuros para implementar acciones preventivas frente a los mismos.
4. Entender la significancia de detectar errores en etapas tempranas, para comprender la necesidad acciones preventivas.

5. Tomar acciones de revisiones previas, para evitar la propagación de errores y defectos.

N5L2: Automatización.

1. Analizar áreas o actividades dentro del proceso de prueba que puedan ser automatizadas.
2. Analizar las posibilidades de automatizar una determinada área o actividades de prueba.
3. Analizar la viabilidad de la automatización de alguna actividad.
4. Plantear objetivos de la automatización.
5. Definir actividades para desarrollar herramientas de automatización.
6. Definir responsable o responsables de desarrollar las herramientas de automatización de pruebas.
7. Analizar resultados de la herramienta de automatización.
8. Determinar la efectividad de las herramientas de automatización.

N5L3: Reuso de material para las pruebas.

1. Establecer un repositorio, accesible para los miembros del equipo donde guardar y consultar el histórico de las pruebas ejecutadas y los resultados de las mismas.
2. Administrar un repositorio de plantillas que permitan llevar un proceso de pruebas ordenado.

N5L4: Mejora continua.

1. Tener claro el enfoque que la organización tiene frente a las pruebas.
2. Mentalizar el cambio y mejora continua para el crecimiento profesional y del proceso en sí.
3. Búsqueda constante de puntos de mejora del proceso.

4. Perfeccionar habilidades y técnicas de pruebas, en base a la experiencia y preparación técnica.
5. Inquietud por siempre buscar aportar un criterio de mejora.

3.4. Planteamiento de un Proceso de Evaluación para el cumplimiento del Marco de Referencia de Calidad.

La evaluación del cumplimiento del Marco de Referencia de Calidad, pretende determinar que tan bien, la organización ha adoptado los procesos y lineamientos planteados por el mismo.

La primera evaluación se recomienda efectuarla al menos tres meses después de haber alcanzado y trabajo bajo los lineamientos planteados en el Nivel 3 de Madurez, ya que así se va a tener un cierto nivel de experiencia y sobre todo se va a tener fundamentos sobre los cuales evaluar la aplicación adecuada del Marco de Referencia.

La evaluación puede comprender un período de unos tres meses, donde de igual manera se va a recopilar información sobre la ejecución del Marco de Referencia, así el proceso de mejora del proceso de pruebas va a basar sus acciones en datos históricos y datos de ejecución, resultantes de la aplicación del Marco al entorno de trabajo.

Posteriormente las evaluaciones se las debe efectuar regularmente, dependiendo del trabajo y desarrollo del equipo, con el fin de tener un control y monitoreo constante del proceso y de esta manera cumplir con el fundamento de mejora continua.

3.4.1. Consideraciones para evaluar el cumplimiento del Marco de Referencia de calidad

Para el Nivel 2 de Madurez, la medición dependerá del equipo de trabajo y del encargado, donde en conjunto determinen si los requisitos, criterios y lineamientos planteados para este nivel han sido alcanzados y adoptados a las prácticas diarias del equipo en sí.

Si después de realizar una autoevaluación al trabajo del equipo, se determina que se ha logrado adoptar las prácticas planteadas para este nivel, de manera que estas se ejecuten con naturalidad por el equipo en los proyectos que trabajen, se puede determinar el equipo ha alcanzado un nivel de madurez adecuado capaz de trabajar sobre un proceso definido, medido y administrado.

Para los niveles de Madurez 3, 4 y 5, se deben considerar factores como la eficiencia del desempeño que el marco de referencia haya tenido a lo largo de su utilización, es decir se deben evaluar factores que permitan determinar el comportamiento del Marco de Referencia frente a las necesidades y labores propias, planteadas por la metodología ágil Scrum, a las cuales se busca aportar con el presente Marco de Referencia.

Otra consideración que se debe tomar en cuenta al evaluar el Marco de Referencia es la usabilidad que tenga dentro de las actividades de la organización, tan integrado está el marco de referencia a la organización y al proceso de pruebas, cómo facilita dicho marco a mejorar el aprendizaje y mejoramiento de las habilidades individuales y grupales, y de igual manera considerar que tan amigable y fácil de aplicar es el marco de referencia para los miembros del equipo.

Para cumplir la evaluación, se puede emplear varias herramientas, como son: encuestas, entrevistas y conversatorios, entre otros, que permitan determinar el nivel de aporte que el Marco de Referencia ha brindado para mejorar las actividades de pruebas y apoyar el desarrollo de la metodología ágil Scrum.

Para aportar a la evaluación, se propone la plantilla denominada *Factores de Mejora para el Proceso*[60], donde en una serie de apartados se espera recopilar los factores que, dentro del desarrollo y ejecución del marco, se hagan evidentes como necesarios de aplicar alguna actividad correctiva, preventiva o de

[60] Factores Para a Mejora del Proceso

regulación, con el fin de mejorar o pulir la implementación y ejecución del marco de referencia como aporte a la organización que desarrolle productos de software bajo los lineamientos de la metodología Scrum.

3.4.2. Inicio de la Evaluación del Marco de Referencia.

1. Evaluar datos históricos de proceso de pruebas ejecutados. (de haberlos)
2. Analizar factores a mejorar que estén siendo trabajados. (de haberlos)
3. Definir el período de evaluación del Marco de Referencia.
4. Definir criterios de medición.
5. Definir objetivos de los criterios de medición.
6. Definir fórmulas para medir los criterios.
7. Documentar los planteamientos para la evaluación

3.4.3. Durante la Evaluación del Marco de Referencia.

1. De acuerdo a los planteamientos definidos para la evaluación.
2. Levantar datos sobre los criterios de medición definidos.
3. Documentar la información obtenida.
4. Actualizar periódicamente la información que se genera gracias a la evaluación

3.4.4. Finalizar la Evaluación del Marco de Referencia.

1. De acuerdo a la información obtenida durante el período de evaluación del Marco de Referencia.
2. Analizar los resultados obtenidos de la evaluación.
3. Plantear acciones reactivas frente a los factores que necesiten ser trabajados.
4. Plantear acciones correctivas frente a los factores que necesiten ser rectificados.
5. Plantear acciones preventivas frente a los factores que necesiten ser mitigados.

Capítulo. V: CONCLUSIONES Y RECOMENDACIONES

En el presente capitulo, se describen las conclusiones a las que se ha llegado tras el desarrollo de la propuesta de Marco de Referencia de Calidad para la Metodología de Desarrollo Ágil Scrum, además se describen ciertas recomendaciones que se han considerado importantes para el óptimo desarrollo de la propuesta antes mencionada.

5.1. Conclusiones.

Después de haber desarrollado la propuesta de Marco de Referencia de Calidad para la Metodología de Desarrollo Ágil Scrum, a modo de conclusiones se establecen los siguientes criterios:

A. La administración efectiva de un proceso de pruebas, puede generar que los resultados de las pruebas ejecutadas sean de mayor validez y cercanos a la realidad esperada del proceso, donde un alto nivel de calidad del producto sea la insignia del mismo.

B. Un análisis previo de los requerimientos de un objeto de prueba, aporta en poder determinar si los mismos están bien definidos o es necesario un mayor detalle, a fin de conocer de manera previa las características funcionales propias del objeto que necesitan ser evaluadas.

C. Documentar las pruebas efectuadas y los resultados obtenidos, representan un significante ahorro de tiempo y recurso, ya que teniendo un fundamento se puede tener un mejor punto de partida para las pruebas y un mejor entendimiento del objeto de prueba que se está evaluando.

D. La planificación previa, antes de comenzar un proceso de pruebas, permite tener un panorama más claro sobre cuál es el objeto de prueba a evaluar y las características que en sí mismo posea, a fin de generar pruebas que cubran de mejor manera estas características y puedan solventar de mejor manera los requerimientos.

E. Registrar y documentar los requerimientos funcionales de un objeto de prueba, permite obtener una base sobre la cual fundamentar que pruebas se deben ejecutar, el porqué de las mismas y aún más importante determinar si es necesario definir requerimientos no

funcionales del objeto que deban ser evaluados para la total aprobación del mismo después del proceso de pruebas.

F. El generar documentación del proceso de pruebas, garantiza que el mismo pueda ser repetible en futuros proyectos, indistintamente del miembro del equipo que lo ejecute. La documentación desarrollada por el proceso de pruebas sirve de base de conocimiento de técnicas, herramientas y características de pruebas que aporten a una mejora de las evaluaciones.

G. Los resultados obtenidos de un proceso de pruebas, son la evidencia de los procedimientos empleados en la evaluación de uno o varios objetos de pruebas, es por ello que la documentación y registro de dichos resultados sirve para un futuro análisis de los mismos y poder tomar decisiones en futuros proyectos.

H. Reportar incidencias encontradas durante el proceso de pruebas permite, de manera efectiva, solucionarlas a fin de que las mismas no generen un mayor desperfecto. Además, reportarlas mediante un artefacto, deja en evidencia su existencia para poder tomar acciones sobre estas en futuros proyectos.

I. Basándose en las incidencias reportadas, analizar las mismas, ayuda a identificar incidencias repetitivas, sobre las cuales se deben tomar acciones a fin de que se encuentre la causa de ellas y poder prevenirlas a futuro.

J. Estructurar la prevención de errores sobre la detección de los mismos, ayuda a que errores comunes no se vuelvan a repetir y que las acciones tanto de desarrollo como de pruebas sean enfocadas a mitigar las causas de los fallos en etapas tempranas del ciclo de vida.

K. Consolidar un proceso de pruebas, permite que las actividades relacionadas a la evaluación y control de calidad de un objeto de pruebas sean repetibles, medibles, profesionales y mejorables a todo nivel, con ello se pueda garantizar le repetición del nivel de calidad en futuros proyectos.

L. Como base para definir un proceso de pruebas, está el consolidar un equipo fuerte de pruebas, el cual tenga un enfoque en la calidad, mejora continua y crecimiento

profesional. Un equipo fuerte de pruebas es el que ha sido preparado técnicamente y ha sido incentivado a compromisos personales y con el equipo, de entregar lo mejor de sus habilidades en pro del beneficio colectivo.

M. Plantear mejoras del proceso de pruebas, en base a los resultados del análisis del desempeño del mismo, ayuda a que las mejoras sean enfocadas en consolidar y mejorar el equipo de pruebas y su labor dentro de la organización. Con la mentalidad de trabajar bajo la mejora continua.

5.2. Recomendaciones.

A. Motivar al personal para el seguimiento del curso de preparación de ISTQB[61] para posterior optar por la certificación en ISTQB Foundation[62]. Con el cual se llega a adquirir lineamientos bases sobre el proceso de pruebas, al igual que fundamentos teóricos sobre técnicas y consideraciones a evaluar para un proceso de pruebas eficiente.

B. Definir un proceso de pruebas que contemple la búsqueda de la mejora continua en un esquema administrable de procesos, lineamientos y procedimientos que aseguren un alto nivel de calidad repetible en otros proyectos.

C. Plantear acciones y estrategias para definir adecuadamente los requerimientos funcionales de los objetos de prueba a evaluar y del proyecto en sí. A poder plantear pruebas que cubran dichos requerimientos de la mejor manera.

D. Documentar los procedimientos de las pruebas ejecutadas y los resultados de las mismas, con la finalidad de llevar un control del trabajo realizado y del cumplimiento de Marco de Referencia de Calidad.

E. Planificar y definir el proceso de pruebas a ejecutar, previo al inicio del mismo, para prevenir la posible falta de detalles en la definición en los objetos de pruebas y preparar el entorno de prueba adecuadamente en base a los requerimientos para la ejecución de pruebas de manera óptima.

[61] (International Software Testing Qualifications Board, 2016)
[62] http://www.istqb.org/certification-path-root/why-istqb-certification/features.html

F. Emplear cuanto más detalle sea posible para documentar el proceso de calidad y los resultados del mismo a fin de poder recopilar información útil y reutilizable para futuros proyectos.

G. Se recomienda utilizar las plantillas propuestas en el presente trabajo, de manera que las mismas se adapten a las necesidades de la organización y contribuyan de mejor manera al desarrollo del proceso de pruebas interno.

H. Incentivar a una constante mejor y capacitar al personal del equipo de pruebas, a fin de motivarles al crecimiento profesional, adquisición y mejora de nuevas habilidades que generen un cambio positivo en el desempeño del equipo.

I. Mantener en un lugar accesible a todo el grupo, toda la información disponible de proyectos trabajados anteriormente con fin de que los mismos puedan acceder a dicha información para tener una referencia del proyecto y además puedan analizar los casos de prueba para desarrollar nuevos planes de prueba.

J. Posterior a la reunión de Planificación de Sprint, donde se definan las funcionalidades a trabajar y requerimientos para el proyecto, se recomienda realizar un análisis de impacto de los cambios requeridos para definir de manera más precisa, el alcance de los cambios y funcionalidades requeridas.

K. Mantener conversaciones en grupo para clarificar cualquier duda y reafirmar el correcto entendimiento de los objetivos del proyecto en ejecución y de los requerimientos que se deben desarrollar durante el mismo.

L. Realizar reuniones con el equipo de pruebas a fin de compartir y socializar los conocimientos, inquietudes y habilidades que puedan generar un enriquecimiento del equipo.

DICCIONARIO DE TÉRMINOS

Núm.	Termino	Significado
1	SCRUM	Scrum es un marco de gestión para el desarrollo incremental de productos, valiéndose de uno o más equipos multi-funcionales, auto-organizados, de aproximadamente siete personas cada uno. Proporciona una estructura de roles, reuniones, reglas y artefactos. Los equipos son los responsables de crear y adaptar sus procesos dentro de este marco. Scrum utiliza iteraciones de longitud fija que se denominan Sprints, que son típicamente de dos semanas a 30 días de duración. Los equipos Scrum intentan generar un incremento de producto potencialmente entregable (debidamente probado) en cada iteración. [63]
2	PRUEBAS (Testing)	Cem Kaner define el testing como una investigación técnica de un producto bajo prueba con el fin de brindar información relativa a la calidad del software, a los diferentes actores involucrados en un proyecto. (Muller, 1989)[64]
3	PRODUCT BACKLOG	Lista ordenada de funcionalidad deseada que es visible para todos los stakeholders donde cualquier stakeholder, incluido el equipo puede agregar ítems. Esta lista es constantemente re-priorizado por el Product Owner.

[63] James, M, Cp. Cit, p 01.
[64] Müller, L. (1989). *CES - Centro de Ensayos de Software.* Obtenido de http://www.ces.com.uy/

4	PRODUCT OWNER	Única persona responsable de maximizar el retorno de la inversión (ROI) del esfuerzo de desarrollo. Es quien es responsable de la visión del producto. Constantemente re-prioriza el Backlog del Producto, ajustando las expectativas a largo plazo, como los planes de liberaciones y además es quien acepta o rechaza cada incremento del producto considerando los intereses de los stakeholders[65]
5	EQUIPO SCRUM	Multifuncional (incluye miembros con habilidades de testing y a menudo otros no llamados tradicionalmente desarrolladores: analistas de negocio, expertos de dominio, etc.). Auto-organizado/auto-gestionado, sin roles asignados externamente[66]
6	SCRUM MASTER	Es el facilitador del proceso de Scrum, quien ayuda a resolver los impedimentos que puedan afectar al equipo Scrum con el fin de crer un ambiente propicio para la auto-organización del equipo[67]

[65] James, M, Cp. Cit, p 01.
[66] Ibídem, p 01.
[67] James, M, Cp. Cit, p 01.

7	PLANIFICACIÓN DE SPRINT (Sprint Planning)	Al comienzo de cada Sprint, el Product Owner y el equipo tienen una Reunión de Planificación del Sprint donde negocian qué ítems del Backlog del Producto intentarán convertir en producto funcionando durante el Sprint. El Product Owner es el responsable de declarar cuáles son los ítems más importantes para el negocio. El equipo es responsable de seleccionar la cantidad de trabajo que cree que podrán realizar sin acumular deuda técnica. El equipo "toma" el trabajo desde el Product Backlog hacia el Sprint Backlog.[68]
8	INGENIERO EN CALIDAD (Tester)	Un Ingeniero en Calidad investiga un producto de software con el objetivo de obtener información acerca de su calidad y del valor que representa para quienes lo utilizan. El Ingeniero en Calidad participa de todas las etapas del proceso de desarrollo de software, colaborando para asegurar la máxima calidad del producto. Su perfil conjuga un conjunto de habilidades con el conocimiento del negocio, de la aplicación bajo prueba y de cómo planificar, diseñar, ejecutar y administrar las pruebas. (Muller, 1989) [69]

[68] James, M, Cp. Cit, p 02.
[69] Müller, L. (1989). *CES - Centro de Ensayos de Software.* Obtenido de http://www.ces.com.uy/

9	REUNIÓN DE SINCRONIZACIÓN (Stand Up Meetings)	Cada día el equipo realiza una reunión de sincronización (15 minutos máximo). Cada miembro del equipo inspecciona el trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para poder hacer las adaptaciones necesarias que permitan cumplir con el compromiso adquirido. (Aguiles.org, n.d.)
10	DEMOSTRACIÓN (Show and Tell)	El equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo. En función de los resultados mostrados y de los cambios que haya habido en el contexto del proyecto, el cliente realiza las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, re planificando el proyecto.
11	RETROSPECTIVA (Retrospective)	El equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad. El Facilitador se encargará de ir eliminando los obstáculos identificados.

ÍNDICE DE ILUSTRACIONES

ANEXOS

PLANTILLAS

PLAN MAESTRO DE PRUEBAS

PLAN MAESTRO DE PRUEBAS

Control de cambios

Introducción

Nombre del Proyecto:	Versión:

Datos Informativos del Proyecto

Fecha de Inicio:	Fecha de Finalización:

Descripción:

Alcance:

Requerimientos Funcionales:

Requerimientos No Funcionales:

Antecedentes	Dependencias:

Cronograma:

Consideraciones para la prueba:

Componentes del Objeto de Prueba:

Tipo de Objeto de Prueba:

Prioridad del Objeto de Prueba:

Recursos necesarios:

Responsabilidades:

Riesgos:

Consideraciones de la Prueba:

Información del Plan Maestro de Prueba

Proceso: Desarrollo / Pruebas

Actividad: Evaluación de Requisitos

Actividad: Diseño de Pruebas

Actividad: Prueba de Implementación

Actividad: Ejecución y revisión de pruebas.

Actividad: Pruebas de Integración

Actividad: Pruebas de Regresión

Proceso: Operación

Actividad: Prueba de funcionamiento

Actividad: Prueba de Carga

Proceso: Mantenimiento

Actividad: Prueba de mantenimiento

Requisitos de documentación para la prueba:

Requisitos administrativos para la prueba:

Requisitos de reporte de resultados de prueba:

Informe de Resultados del Plan Maestro de Prueba

Descripción General del proceso de Pruebas:

Resumen de las pruebas individuales:

Análisis de Incidencias:

Acciones sugeridas:

Resumen de resultados de prueba por actividad:

Proceso: Desarrollo / Pruebas

Actividad: Evaluación de Requisitos

Actividad: Diseño de Pruebas

Actividad: Prueba de Implementación

Actividad: Ejecución y revisión de pruebas.

Actividad: Pruebas de Integración

Actividad: Pruebas de Regresión

Proceso: Operación

Actividad: Prueba de funcionamiento

Actividad: Prueba de Carga

Proceso: Mantenimiento

Actividad: Prueba de mantenimiento

ESQUEMA DEL DISEÑO DE PRUEBAS

ESQUEMA DEL DISEÑO DE PRUEBAS

Control de cambios

Introducción

Identificación del (de los) Objeto (s) de Prueba:

Alcance del (de los) Objeto (s) de Prueba:

Referencias del (de los) Objeto (s) de Prueba:

Detalle del Diseño de Pruebas

Objeto(s) a ser probado (s):

Criterios de entrada:

Criterios de salida:

Entregables:

ESQUEMA DE CASOS DE PRUEBA

ESQUEMA DE CASOS DE PRUEBA

***Se recomienda generar un documento de Esquema de Casos de Prueba por cada objeto de prueba.**

Control de cambios

Introducción*

Descripción del Caso de Prueba:

Alcance:

Fecha de Inicio: **Fecha de Finalización:**

Estado: **Responsable**: **Prioridad:**

Detalles*

Objetivos:

Entrada(s):	**Salida(s):**

Requerimientos de entorno:

Requisitos de procedimiento especiales:

Interdependencias:

Flujo Normal*

Actor (Acción):	Sistema (Resultado):

Flujo Alterno*

Actor (Acción):	Sistema (Resultado):

LISTA DE VERIFICACIÓN

LISTA DE VERIFICACIÓN

Introducción

Nombre del Proyecto:
Fecha Inicio Pruebas:
Probador:
Descripción de la Prueba:

Verificaciones

#	CRITERIOS	SI	NO	COMENTARIOS
1.				
2.				
3.				
4.				
5.				
6.				
7.				
8.				
9.				
10.				
11.				
12.				
13.				
14.				
15.				
16.				
17.				
18.				
19.				
20.				

Fecha Finalización Pruebas:
Probador:

Detalle de la prueba

Resultados de Pruebas:

Comentarios de las Pruebas:

Casos de Pruebas:

REPORTE DE INCIDENCIAS

REPORTE DE INCIDENCIAS

Control de cambios

Introducción

Descripción de la Incidencia:

Fecha Incidencia Encontrada:

Responsable: **Prioridad:**

Detalles

Impacto – Áreas Afectadas:

Posibles causas de la Incidencia:

Acciones Correctivas:

Estado de la Incidencia:

Conclusiones:

REPORTE DE OBJETO DE PRUEBA

REPORTE DE OBJETO DE PRUEBA

Control de cambios

Introducción

Descripción de la Prueba:

Fecha de la Prueba:

Responsable: **Prioridad:**

Detalles

Detalles de los resultados:

Anomalías Encontradas:

Estado de la Prueba:

Conclusiones:

CATÁLOGO DE PRUEBAS

Índice de Pruebas

No. 1	**Proyecto:**	**Objeto de Prueba:**
Comentarios:		**Fecha Inicio:** **Fecha Finalización:**
Ubicación / URL:		
No. 2	**Proyecto:**	**Objeto de Prueba:**
Comentarios:		**Fecha Inicio:** **Fecha Finalización:**
Ubicación / URL:		
No. 3	**Proyecto:**	**Objeto de Prueba:**
Comentarios:		**Fecha Inicio:** **Fecha Finalización:**
Ubicación / URL:		
No. 4	**Proyecto:**	**Objeto de Prueba:**
Comentarios:		**Fecha Inicio:** **Fecha Finalización:**
Ubicación / URL:		
No. 5	**Proyecto:**	**Objeto de Prueba:**
Comentarios:		**Fecha Inicio:** **Fecha Finalización:**
Ubicación / URL:		
No. 6	**Proyecto:**	**Objeto de Prueba:**
Comentarios:		**Fecha Inicio:** **Fecha Finalización:**
Ubicación / URL:		

FACTORES PARA LA MEJORA DEL PROCESO

FACTORES PARA LA MEJORA DEL PROCESO

Fecha Inicio Mejora:

Proyecto:

Componentes de Mejora

Factores Fuertes:

Factores Débiles:

	FACTORES FUERTES	FACTORES DÉBILES
INDIVIDUALES		
	FORTALEZAS	DEBILIDADES
	OPORTUNIDADES	AMENAZAS
GRUPO		

Proceso de Mejora

Factores a Mejorar:

Acciones a Ejecutar:

Criterios de Evaluación:

Número de Factores a mejorar: **Número de Factores mejorados:**

Porcentaje Factores mejorados (6*100%)/5**:**

Resultados de Proceso de Mejora

REFERENCIAS

Agile For All. (2016). *Agile For All - Resources*. Retrieved from Agile For All: http://agileforall.com/resources/introduction-to-agile/

Aguiles.org, P. (n.d.). *Proyectos Aguiles*. Retrieved from http://proyectosagiles.org/que-es-scrum/

American National Standards Institute. (2016). *American National Standards Institute*. Retrieved from https://www.ansi.org/

Arianespace. (2015). *Arianespace - Services & Solutions*. Retrieved from http://www.arianespace.com/launch-services-ariane5/ariane-5-intro.asp

Asociación Española para la Calidad. (2016). *Asociación Española para la Calidad*. Retrieved from http://www.aec.es/web/guest/centro-conocimiento/modelos-de-calidad

Atom - SPICE/ISO /IEC 15504. (2012, Mayo 09). *SPICE/ISO /IEC 15504*. Retrieved from http://seispice.blogspot.com/: http://seispice.blogspot.com/2012/05/spiceiso-iec-15504-norma-spiceiso-iec.html

Carnegie Mellon. (2010, Enero 15). *Software Engineering Institute*. Retrieved from Team Software Process: http://resources.sei.cmu.edu/asset_files/Brochure/2010_015_001_72817.pdf

CEPPE. (n.d.). *CEPPE – Comité para la Evaluación de Programas de Pedagogía y Educación*. Retrieved from http://www.ceppe.org.mx/acreditacion/marco-de-referencia/

Das, Ranjna. (2015). *Define International Quality Standards*. Retrieved from http://www.ehow.com/: http://www.ehow.com/facts_7366023_define-international-quality-standards.html

Fourman, M. (2001, 10 22). *Rational Unified Process Process Description and Workflows.* Retrieved from http://www.dcs.ed.ac.uk/teaching/cs2/online/Lectures/CS2Ah/SoftEng/se02-slides.PDF

Gómez Arenas, L., & Ramos, C. (2013, Julio 20). Paquete de Despliegue - Integración y Pruebas - Perfin Básico. Colombia.

http://proyectosagiles.org/que-es-scrum/. (n.d.). Diagrama Proceso Scrum. *Qué es SCRUM.*

Ibáñez, Á. (2014, Junio 04). *RTVE*. Retrieved from http://www.rtve.es/noticias/20140604/error-software-convirtio-lanzamiento-espacial-carisimos-fuegos-artificiales/948262.shtml

International Software Testing Qualifications Board. (2016). Retrieved from ISTQB.ORG: http://www.istqb.org/

ISO 25000. (2015). *ISO 25000 Calidad del Producto Software*. Retrieved from http://iso25000.com/index.php/normas-iso-25000/iso-25010?limit=3&limitstart=0

James, M. (2010-2012). *Scrum Reference Card - Español.* Retrieved from http://scrumreferencecard.com/

Jiménez, C. (15 de Octubre de 2012). *SlideShare*. Obtenido de http://es.slideshare.net/cejird/clase-5-marco-de-referencia

Muller, L. (1989). *CES - Centro de Ensayos de Software.* Retrieved from http://www.ces.com.uy/

Paez Bernal, J. E. (2009, Octubre 28). *http://lasnormasiso15504122079126.blogspot.com*. Retrieved from http://www.blogspot.com: http://lasnormasiso15504122079126.blogspot.com/2009/10/las-normas-isoiec-15504-12207-9126.html

Palacio, Juan; Sinopsis de los modelos SW-CMM y CMMi. (2006, Abril). *Sinopsis de los modelos SW-CMM y CMMi.* Retrieved from Navegapolis.com: http://navegapolis.com/files/sinopsis_cmm.pdf

Project Management Institure . (2016). *Project Management Institute - America Latina.* Retrieved from Project Management Institure: http://americalatina.pmi.org/latam/pmbokguideandstandards/whatisastandar.aspx

Pyhäjärvi, M. (2004, Noviembre 31). SPICE – International Standard for Software Process Assessment. UNIVESITY OF HELSINKI, Department of Computer Science, Seminar on Quality Models for Software Engineering, Helsinki.

Rational Software- The Software Development Company. (1998). Rational Unified Process - Best Practices for Software Development Teams. Cupertino, CA, USA.

Romero, C. I. (2001, Mayo 15). *Colección de Tesis Digitales.* Retrieved from Universidad de las Américas Puebla: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/garcia_r_ci/indice.html

Scrumguides.org. (n.d.). *The History of Scrum.* Retrieved from http://www.scrumguides.org/history.html

Software & Systems Engineering Standards Committee. (2008, Julio 18). IEEE Standard for Software and System Test Documentation - IEEE Std 829-2008. New York, New York, U.S.A.

Software & Systems Engineering Standards Committee. (2008, Agosto 15). IEEE Standard for Software Review - IEEE Std 1028-2008. New York, New York, U.S.A.

Software Engineering Institute. (2007). Retrieved from Software Engineering Institute - Carnegie Mellon University: http://www.sei.cmu.edu/library/assets/cmmi-overview071.pdf

TMMi Foundation. (2012). *Test Maturity Model integration (TMMi) Release 1.0.* Retrieved from www.tmmi.org: http://www.tmmi.org/pdf/TMMi.Framework.pdf

Xplore, I. (2012, Marzo 10). *http://dis.unal.edu.co/.* Retrieved from Universidad Nacional de Colombia: http://dis.unal.edu.co/~icasta/ggs/Documentos/Normas/610-12-1990.pdf