

*Universidad de las Ciencias Informáticas*



*Facultad 9*

*Trabajo de Diploma para Optar por el Título de Ingeniero en  
Ciencias Informáticas.*

*Título:* Propuesta de adopción del framework TMMi como modelo de referencia para el proceso de pruebas en los proyectos productivos de la facultad 9.

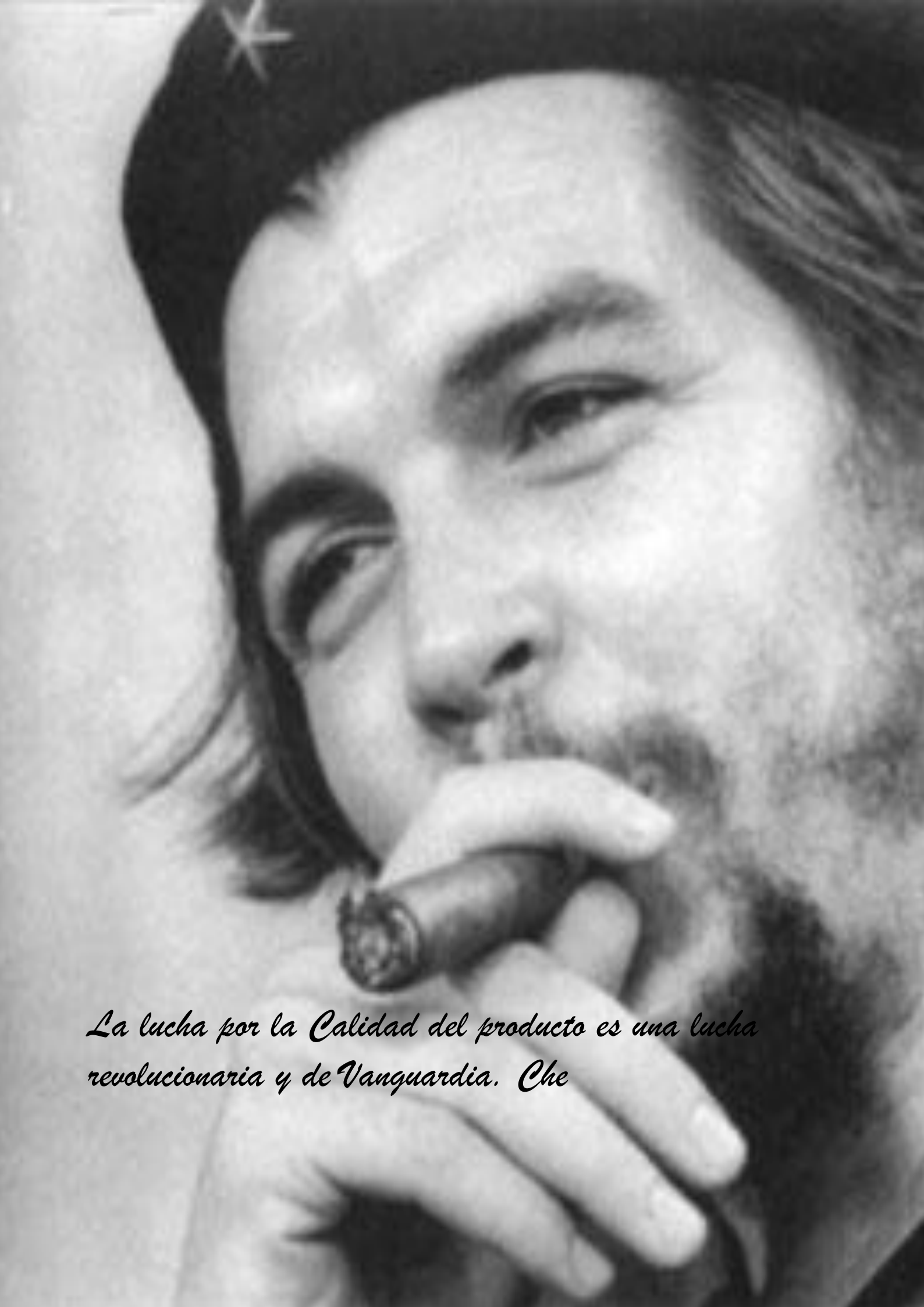
*Autor:* Hector Manuel Blanco Meriño.

*Tutor:* Ing. Jorge Infante Osorio.

*Ciudad de la Habana, Junio 2009*

*"Año del 50 Aniversario de*

*Triunfo de la Revolución"*



*La lucha por la Calidad del producto es una lucha revolucionaria y de Vanguardia. Che*

# Dedicatoria

---

## **Dedicatoria:**

*Le dedico este trabajo a mi madre por ser tan especial y darme todo en esta vida.*

*A mis abuelos Orlando, Magdalena y Eloida.*

*A mi padrastro Cristóbal Muztelier Cruz por haberme criado y darme el amor de un padre.*

*A mis hermanas Yamila y Yamileisis por ser mis tesoros.*

*A mi sobrina Liz Jennifer.*

*A mi papá Hector Antonio.*

*A mis tíos Urbano y Lalo por enseñarme la vida.*

*A toda mi familia por darme todo su apoyo cuando siempre lo he necesitado.*



# Agradecimiento

---

## *Agradecimientos:*

*A mis profesores de esta universidad por haberme enseñado tanto.*

*Al profesor Larrude que siempre confió en mí.*

*A mi Facultad 9 que siempre lo di todo por ella.*

*A todos mis compañeros durante estos cinco largos años.*

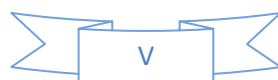
*A Raúl por haberme ayudado con la realización de este trabajo.*

*A los de mi proyecto por apoyarme siempre.*

## Resumen

Hoy en día las empresas productoras de software buscan alternativas para hacer más eficientes sus desarrollos, en aras de garantizar la calidad y la satisfacción del usuario, potenciando así un aumento de la cultura hacia la excelencia, que de como resultado una mejor producción. Para lograr una mejor producción y que las soluciones informáticas tenga la calidad que espera el cliente, la industria del software se ha centrado mucho en la mejora de sus procesos de desarrollo utilizando para esto el Modelo de la Capacidad de Madurez Integrado (CMMi, sus siglas en inglés). Diversas organizaciones actualmente se encuentran enfrascadas en la mejoras del proceso de pruebas, sin embargo CMMI no le proporciona el soporte adecuado para realizar dicha mejora. La integración de CMMI con otros modelos de calidad es muy costosa debido a la incompatibilidad que existe entre ellos. En este trabajo se propone que se adopte el framework Modelo Integrado de Madurez de Prueba (TMMi, sus siglas en inglés) ya que el mismo se posiciona como un complemento para el CMMi y puede ser utilizado como un modelo de referencia para la mejora del proceso de pruebas, además su estructura es similar a la del CMMI lo que facilita una mayor comprensión ya que muchas personas utilizan CMMI para la mejora de sus procesos.

**Palabras claves:** Calidad, Modelo de Referencia, Mejora del proceso de Prueba, TMMi.



## Índice

Introducción .....	9
Capítulo 1: Fundamentación Teórica y Tendencias usadas para la solución del problema.....	16
1.1 Introducción.....	16
1.2. Principales conceptos asociados al tema. ....	16
1.2.1 Calidad.....	16
1.2.2 Calidad del Software. ....	17
1.2.3 Pruebas de Software.....	18
1.2.4 Proceso de desarrollo de software. ....	20
1.2.5 Modelos de procesos en el desarrollo de software.....	22
1.2.6 Mejora de proceso en el desarrollo de software.....	23
1.2.7 Modelo de referencia para el proceso de pruebas.....	24
1.3. Principales modelos usados en el proceso de desarrollo de software. ....	24
1.3.1 Características del Modelo de Capacidad de Madurez (Capability Maturity Model – CMM). ....	24
1.3.2 Características del Modelo de Capacidad de Madurez Integrado (Capability Maturity Model Integration-CMMi). ....	27
1.4 Proceso Unificado de Desarrollo de Software.....	30
1.5 Objeto de Estudio.....	32
1.5.1 Descripción General del Objeto de estudio. ....	32
1.5.2 Situación Problemática. ....	33
1.5.3 Análisis de otras soluciones existentes.....	34
1.6 Conclusiones Parciales.....	34

---

Capítulo 2: Solución Propuesta.....	36
2.1 Introducción.....	36
2.2 Caracterización del Modelo Integrado de Madurez de Prueba.....	36
2.2.1 Antecedentes e Historia.....	36
2.2.2 Fuentes.....	37
2.2.3 Ámbito de aplicación de TMMi.....	37
2.2.3.1 Ingeniería de Sistemas y de Software.....	38
2.2.3.2 Nivel de Prueba.....	38
2.2.3.3 TMMi y CMMi.....	38
2.2.3.4 Evaluaciones.....	38
2.2.4 Niveles de Madurez.....	39
2.2.4.1 Nivel Inicial.....	39
2.2.4.2 Nivel 2 Gestionado.....	40
2.2.4.3 Nivel 3 Definido.....	41
2.2.4.4 Nivel 4 Gestión y Medición.....	42
2.2.4.5 Nivel 5 Optimización.....	42
2.2.5 Estructura del TMMi.....	43
2.2.5.1 Componentes requeridos, esperados e Informativos.....	43
2.2.5.2 Componentes del TMMi.....	44
2.3 Caracterización de las Áreas de Proceso.....	46
2.4 Propuesta de Prácticas y Metas específicas para alcanzar el nivel 2 de TMMi. Área de proceso Política de Prueba y Estrategia.....	47
2.4.1 Propósito.....	48
2.4.2 Alcance.....	48

2.4.3 Prácticas y metas específicas. ....	48
2.5 Estrategia General para aplicar el framework TMMi a los proyecto de la Facultad 9.....	50
2.6 Propuesta de capacitación del personal. ....	53
2.5.1 Curso Optativo Introducción al TMMi. ....	54
2.7 Conclusiones Parciales.....	57
Capítulo 3: Evaluación de la propuesta.....	58
3.1 Introducción.....	58
3.2 Método de Evaluación de Expertos. ....	58
3.2.1 Elaboración del objetivo.....	59
3.2.2 Selección de los expertos.....	59
3.2.3 Elaboración del cuestionario.....	59
3.2.4 Elección de la metodología. ....	59
3.2.5 Ejecución de la metodología.....	60
3.2.6 Procesamiento de la información.....	60
3.3 Análisis de beneficios y costo. ....	64
3.4 Conclusiones Parciales.....	66
Conclusiones Generales .....	67
Recomendaciones.....	68
Referencias Bibliográficas.....	69
Bibliografía Consultada.....	72
Glosario de Términos.....	74
Anexos.....	75



## **Introducción**

El mundo ha entrado en una etapa de rápidos avances científicos y tecnológicos que dan lugar a un uso más excesivo de nuevas tecnologías, especialmente en los campos de la bioingeniería y los nuevos materiales, del procesamiento de la información y automatización de tareas (informática, multimedia, robótica), del transporte (aéreo y de superficie) y de los medios de comunicación (telefonía, redes, Internet...), que proporcionan potentes herramientas para el tratamiento de la información y permiten establecer redes de comunicación que facilitan una veloz circulación de personas, mercancías, dinero e información por todo el planeta. Esto provoca que muchas personas en el mundo hagan un uso masivo de las Tecnologías de la Información y las Comunicaciones (TIC) obligando a desarrollar software que satisfaga la demanda de la sociedad.

Con el acelerado uso de las Tecnologías de la Información y las Comunicaciones (TIC), los problemas van siendo más complejos, esto obliga a buscar nuevas soluciones, nuevos caminos o nuevo paradigmas que solucionen dichos problemas. La solución generalmente incluye el desarrollo de un sistema informático que maneje la gran cantidad de información y la complejidad del problema, el desarrollo de software se ha convertido en una tarea muy compleja que ha sobrepasado en gran medida la habilidad para el mantenimiento de las empresas productoras.

Hoy en día las empresas productoras de software buscan alternativas para hacer más eficientes sus desarrollos, en aras de garantizar la calidad y la satisfacción del usuario, potenciando un aumento de la cultura hacia la excelencia y la administración del desarrollo, que den como resultado una mejor producción y empleo de los recursos para el desarrollo del software.

La calidad de software como disciplina encargada de garantizar el cumplimiento de los requerimientos impuestos por los usuarios para el desarrollo de los sistemas de software se ha convertido en una necesidad de las empresas que se dedican a la producción del software. En los últimos años el tema de la calidad dentro del desarrollo de software ha ido en aumento debido fundamentalmente a la llamada crisis de la industria del desarrollo de software, lo cual en gran medida se ha debido a la falta de implementación de procesos de pruebas que garanticen el producto final.

La industria del software viene realizando grandes esfuerzos por mejorar la calidad de los productos, tarea difícil, ya que el tamaño y la complejidad de los sistemas de software aumentan rápidamente, mientras que los clientes y usuarios son cada vez más exigentes.

A pesar de resultados alentadores con diversos enfoques de mejora de la calidad, la industria del software aún está lejos de lograr los cero defectos.

Para mejorar la calidad de los productos, la industria del software se ha centrado mucho en la mejora de sus procesos de desarrollo. Una directriz que se ha utilizado ampliamente para mejorar los procesos de desarrollo es el Modelo de la Capacidad de madurez (CMM, sus siglas en inglés) y su sucesor el Modelo de la Capacidad de Madurez Integrado (CMMi, sus siglas en inglés). Siendo a menudo estos modelos considerados como el estándar de la industria del software para la mejora de procesos (**ORÉ, 2009**).

El ciclo de vida del software involucra una serie de procesos. El proceso de desarrollo involucra: el análisis de los requisitos del software, el diseño detallado del software, codificación y **prueba del software**. El proceso de soporte incluye: el proceso de aseguramiento de la calidad, proceso de verificación, proceso de validación.

Las pruebas de software, vistas desde el marco de un proceso de desarrollo de software, son los diferentes procesos que se deben realizar durante un desarrollo, con el objetivo de asegurar la calidad, entre otros factores de gran importancia.

Estas, al contrario de lo que muchas personas creen, no se deben dejar para el final de la etapa de desarrollo del software. Las pruebas se deben empezar a realizar desde la misma etapa de análisis y levantamiento de requerimientos, ya que desde un principio se puede incurrirse en malas interpretaciones de las reglas del negocio, lo que finalmente tendrá como consecuencia incongruencia entre lo que el cliente quiere y lo que se ha desarrollado.

A pesar del hecho de que las pruebas de software a menudo cuestan por lo menos el 30 ó 40% del total del proyecto, se presta poca atención a las pruebas en los distintos modelos de mejora de procesos tales como el CMM y el CMMI (**ORÉ, 2009**).

Las ventajas que trae la realización de pruebas, en un desarrollo de software son las siguientes (**CALDAS, 2001**):

- Reducen la posibilidad de agregar defectos al software. Si hay que realizar una adición de características requeridas por el cliente y se ve que ya no funcionan bien algunas de las cosas

que anteriormente servían, se puede inferir la nueva funcionalidad es la que contiene defectos, por lo que no hay necesidad realizar modificaciones a los componentes realizados anteriormente.

- Reducen la posibilidad de encontrar defectos en funcionalidades ya implementadas.
- Las pruebas son buena documentación. Es preferible ver unas pequeñas líneas de "Código de prueba", que son concisas y realmente son fáciles de entender, a revisar línea por línea de código para poder entender que hace determinado componente de software.
- Reducen el costo del cambio. Ya que evitan que se descubran los defectos hasta el final del desarrollo de software.
- Permiten realizar reutilización. Se puede llegar a necesitar la reutilización de determinada funcionalidad en un sistema, debido a fallos de seguridad, rendimiento, o simplemente porque no reunía lo que el cliente esperaba de éste, entonces dada tal situación, las pruebas que a realizar a dicha funcionalidad van a permitir que se vuelva a desarrollar de manera segura, ya que la prueba encierra el criterio de aceptación sobre la funcionalidad, permitiendo de esta forma que se vuelva a desarrollar algo acorde con la especificación.
- Restringe las características a implementar. Muchas veces los programadores pierden tiempo en detalles que la especificación no pedía. Con las pruebas el programador sabe que tiene que programar dicha funcionalidad y también probarla, por lo que se restringe a lo que los diseñadores de las pruebas hayan realizado.
- Hacen que el desarrollo sea más rápido. Ya que a medida que se agregan características al software, las funcionalidades anteriores pueden fallar, pero si se han hecho las pruebas pertinentes a las funcionalidades anteriores, se puede descartar inmediatamente que existan defectos en éstas, por lo que se puede concentrar tranquilamente en la funcionalidad nueva.

La calidad de las pruebas de software vendrá determinada por la calidad del proceso de pruebas que utilice. Así la solución a muchos de los problemas relacionados con las pruebas consiste en tener un proceso de pruebas bien definido, gestionado y controlado, que comience con la fase de definición de requisitos y se desarrolle de forma paralela al proceso de desarrollo del software.

Un proceso de pruebas bien definido implica (**SANZ, 2008**):

- Identificar y establecer los objetivos, políticas y estrategia de pruebas que guiarán el proceso.

- Desarrollar y mantener un plan de pruebas que recoja los aspectos necesarios para la gestión y control de las actividades de pruebas.
- Definir las especificaciones necesarias para generar los casos de pruebas adecuados.
- Identificar y establecer las medidas adecuadas para solucionar los posibles resultados encontrados.

La mejora del proceso de pruebas generará un conjunto de beneficios para la industria del software. Los más destacables son los siguientes (**SANZ, 2008**):

- Se produce un incremento de la satisfacción del cliente al utilizar un software con una calidad de errores inferiores.
- Se incrementa la eficiencia del proceso de desarrollo.
- Se facilita la definición y cumplimiento de los objetivos de calidad.
- Se incrementa la satisfacción de los trabajadores debido a que proporcionan herramientas y recursos apropiados para la realización eficiente del trabajo.

En aras de mejorar la calidad del proceso de pruebas y así mejorar la calidad de los productos nuestra universidad al igual que las demás empresas productoras de software se están preocupando de implantar modelos de calidad que le den soporte al proceso de pruebas, con el objetivo de que sus proyectos sean más predecibles, se reduzcan los riesgos durante el desarrollo y que los mismos tengan la calidad que espera el cliente.

La universidad juega un papel importante en el desarrollo de la Industria Cubana del Software (ICSW), y en la materialización de los proyectos asociados al programa cubano de informatización. Ya que las producciones de software son cada vez cada vez más importantes e imprescindibles para impulsar el desarrollo de la informática en Cuba, las mismas constituyen uno de los principales pilares fundamentales en el acelerado proceso de informatización de la sociedad.

Producir un software de alta calidad hoy en día es una meta importante y no es difícil de alcanzar. Según expresara Ernesto Guevara de la Serna (Che) “la lucha por calidad del producto es una lucha revolucionaria y de vanguardia” (**LLEBREZ y CASTILLO, 2006**).

Nuestra universidad al igual que diferentes empresas del mundo está trabajando con CMMi para la mejora de sus procesos en el desarrollo de software. Según el perfil de madurez mundial de las

organizaciones intensivas en software, desarrollado por el Instituto de Ingenieros de Software (SEI, por sus siglas en inglés), de todas aquellas organizaciones que evaluaron sus procesos al nivel 2 de CMMi para los Procesos de Verificación y Validación, menos del 2% de las organizaciones lograron satisfacer totalmente ese nivel, mientras que las organizaciones que lo lograron satisfacer parcialmente, es decir no en la totalidad de los requisitos marcados por el CMMI, no superan el 4,5% (SANZ, 2008).

Al igual que la universidad la facultad 9 se encuentra interesada en la mejora de su proceso de pruebas pero según una publicación del Dr. Luís Fernández Sanz del departamento de Sistemas Informáticos de la Universidad Europea de Madrid en la Revista Española de Innovación, Calidad e Ingeniería de Software (REICIS, sus siglas) CMMi no proporciona el soporte necesario para la mejora de sus proceso de pruebas. En relación al proceso de pruebas CMMi define 3 áreas de proceso: Integración de Producto, Validación y Verificación; sin embargo no cubren todas las necesidades del proceso de pruebas. El grado de abstracción con respecto a este proceso es muy elevado, lo trata como algo genérico mientras que necesita de una atención más centrada y una definición exhaustiva (SANZ, 2008). Esto trae consigo que nuestra facultad al igual que las demás empresa productoras de software se encuentren en la necesidad de aplicar modelos de calidad que se integren con CMMi y que a la vez mejoren el proceso de pruebas.

Por todo lo expresado anteriormente se llega al siguiente **problema a resolver** de ¿cómo estandarizar un modelo de referencia para el proceso de pruebas en el entorno productivo y metodológico de los proyectos productivos de la facultad 9? Para ello se ha tomado como **objeto de estudio** el Proceso de desarrollo de software en los proyectos productivos de la facultad 9 y el **campo de acción** que abarca este trabajo es el proceso de pruebas como disciplina en los proyectos productivos de la facultad 9.

El **objetivo general** se traduce entonces en: Realizar una propuesta para la aplicación del *framework* Modelo Integrado de Madurez de Prueba (TMMi, sus siglas en inglés), como modelo de referencia para el proceso de pruebas y mejora, dentro del proceso de desarrollo de software de los proyectos productivos de la facultad 9.

Los **objetivos específicos** son:

- Caracterizar el *framework* TMMi.

- Definir una estrategia para aplicar el framework a los proyectos de la Facultad 9.
- Mejorar el proceso de pruebas que contiene el Proceso Unificado de Desarrollo de Software (RUP, sus siglas en inglés) como disciplina de pruebas con la utilización del modelo de referencia que presenta el *framework* TMMi.

Como **posible resultado** es caracterización del Modelo Integrado de Madurez de Prueba que permita mejorar el proceso de prueba que utilizan los proyectos de la facultad 9 y definir una estrategia para aplicar el modelo a los proyectos de la facultad 9.

Para el mismo se ha planteado las siguientes **Hipótesis**:

Si se aplica el modelo de referencia propuesto en TMMi al proceso del desarrollo de software, en la disciplina de pruebas, dentro de los proyectos productivos de la facultad 9 entonces se garantizará una estandarización a nivel de la organización de dicha disciplina potenciando la calidad del proceso de pruebas y de los productos desarrollados.

Para darle cumplimiento a los objetivos trazados y validar la hipótesis de la investigación, se determinó que las tareas a realizar en esta investigación estarían dirigidas a:

1. Valorar el estado del arte los proyectos de la facultad 9 en el proceso de pruebas
2. Valorar el estado del arte de los modelos de calidad existentes.
3. Analizar las tendencias actuales del uso del *framework* TMMi.
4. Caracterizar el *framework* TMMi.
5. Definir una estrategia para aplicar el framework a los proyectos de la facultad 9.
6. Validar la solución propuesta.

En esta investigación se hace uso de los **métodos teóricos y empíricos** debido a que nos permiten realizar una guía de la información que posibilita a su vez tener los datos necesarios para dar respuesta al problema. Dentro de los métodos teóricos se utiliza el método **histórico-lógico** porque este permite conocer sobre el problema a resolver desde sus principales etapas en su trayectoria, la historia de su desarrollo y garantizan el estudio de la estructura del objeto de la investigación junto a la

concepción histórica del mismo. Posibilita saber cuál es la verdadera situación del tema en la actualidad.

El **Analítico-Sintético**: El tener un gran cúmulo de información permitirá resumir y sintetizar dicha información en busca de los objetivos perseguidos.

Dentro de los **métodos empíricos la observación** siempre va a estar presente porque este método permite conocer la realidad mediante percepción del objeto de estudio, es decir, cuando se comienza la investigación se basa en la observación para saber cómo son los problemas existentes, se utilizan **la entrevista** que mediante las misma se conoce que piensan los líderes y los demás integrantes de los proyectos de la facultad 9 relacionado con el objeto de estudio para mediante los problemas que encuentran poder mejóralos y obtener un mejor resultado.

Se requiere tener una **población** que son el conjunto de los 19 proyectos en la Facultad 9. Haciendo uso de un muestreo no probabilístico, usando la técnica de muestreo intencional, se obtuvo una **muestra** de 7 proyectos de la Facultad 9 el proyecto canal informativo para el Ministerio del Poder Popular para la Energía y Petróleo (MENPET, sus siglas) y el del proyecto de Sistema Gestor de Datos Geológicos (SGDG, sus siglas), el proyecto de Plataforma de video web, el proyecto de Monitoreo de TV, ACN v2.0 y el proyecto de Grupo de Desarrollo de Sistemas de Información Geográficas(GSIG, sus siglas).

El trabajo que se presenta a continuación está estructurado en tres capítulos. El Capítulo 1 contempla la fundamentación teórica de esta investigación, en la cual son expuestos los principales conceptos que contribuyen al mejor entendimiento del problema en cuestión, se especifican detalladamente todos los argumentos que esclarecen el objeto de estudio. En el Capítulo 2 se realiza una caracterización detallada del *framework* TMMi y sus beneficios. En el capítulo 3 se realizará la validación de la solución propuesta a través de un panel de expertos. Para finalizar se presentan las conclusiones, las recomendaciones, las referencias bibliográficas, la bibliografía, un glosario de términos y el conjunto de anexos para que quede más claro la explicación que se haya dado anteriormente.

## Capítulo 1

### Capitulo 1: Fundamentación Teórica y Tendencias usadas para la solución del problema.

#### 1.1 Introducción.

En este capítulo se hace referencia a los fundamentos teóricos que sustentan la investigación. Se analiza el estado del arte del tema tratado, a nivel internacional, nacional y de la universidad, sus principales tendencias para solución del problema que se enfrenta. Se hace un estudio de los modelos de calidad existentes y su implicación con el tema.

#### 1.2. Principales conceptos asociados al tema.

##### 1.2.1 Calidad

Desde finales del siglo XIX, con la revolución industrial, muchos fabricantes incipientes de la época quisieron otorgar calidad a sus productos, pero la calidad como tal era entendida en el sentido de otorgar uniformidad al producto final.

A partir de la década de 1930 y hasta 1950 el enfoque de calidad varió gracias al aporte de W. Stewart y E. Deming, ellos desarrollaron controles estadísticos para los procesos de producción, una de las primeras aplicaciones fue hecha en los laboratorios Bell en los Estados Unidos, pero con el surgimiento de la segunda guerra mundial se hizo masivo.

En la década del 80 surge el concepto de aseguramiento de la calidad el cual estaba enfocado a la prevención de los problemas y hacía énfasis en los ciclos completos de la producción pero con la contribución de los grupos funcionales de cada etapa, estos grupos funcionales eran especializados en la calidad, su auge se debió a que obtuvo un apoyo muy fuerte en los sistemas de información, que ya habían aparecido para entonces, y que brindaban una manera más eficiente de gestión. Paralelamente Japón adoptó el concepto de Gestión Total de la Calidad que comprometía a todo el personal de la organización en las tareas de aseguramiento de la calidad.

Con el surgimiento del concepto de Gestión Estratégica de la Calidad se enfocó la calidad como una medida de competitividad de las organizaciones, la globalización de finales del siglo pasado hizo que



# Capítulo 1: Fundamentación Teórica

---

las organizaciones hicieran más énfasis en las necesidades de los diferentes mercados y por consecuencia en las necesidades de sus clientes, y que la calidad del producto fuera determinada por el cumplimiento de estas necesidades.

Actualmente existen muchos conceptos de calidad y todos reflejan como principal enfoque la competitividad para alcanzar las metas trazadas que es ofrecer productos y servicios de calidad y satisfacer las necesidades de los mercados y los clientes.

La Organización Internacional de Estándares (ISO, por sus siglas en inglés) ha publicado hasta hoy varios estándares relacionados con la calidad en general y también de manera particular con la calidad de software, de todas las normas ISO publicadas, la ISO 9000 corresponde al estándar de gestión de calidad ampliamente aceptado y que ha sido la base para otras normas más específicas, la norma ISO 9000 conceptualiza la calidad como “el grado en el que un conjunto de características inherentes cumple con los requisitos” (**ISO9000, 2000**) a su vez para la norma ISO 8402 calidad se define como: “Totalidad de características de un producto que le confieren su aptitud para satisfacer unas necesidades expresadas o implícitas” (**ISO8402, 1994**), debe entenderse que para esta norma la expresión “necesidades expresadas o implícitas” se refiere a que las necesidades pueden ser definidas por un contrato o son inherentes a la funcionalidad del producto (**ESPINOZA y ESPINOZA, 2005**).

Otro estándar muy importante pero aplicable sólo a su región es el Estándar Industrial Japonés (JIS, por sus siglas en inglés) z8110-1981 (**JSI, 1981**) establece el concepto de calidad como “La totalidad de características o rendimiento, que puede ser usado para determinar si un producto cumple o no su aplicación prevista o intencionada” (**ESPINOZA y ESPINOZA, 2005**).

## 1.2.2 Calidad del Software.

La calidad de software comúnmente aceptada por la comunidad científica internacional constituye la principal meta de todas las empresas desarrolladoras de software en el mundo y es el orgullo de las entidades producir un producto con la calidad que espera el cliente y que pueda satisfacer sus necesidades y expectativa esperada.

Un software hecho para ejecutarse una sola vez no requiere el mismo nivel de calidad que un software elaborado para ser explotado durante un largo período. Este necesita ser confiable, mantenible y flexible para disminuir los costos de mantenimiento y perfeccionamiento durante el tiempo de explotación.

# Capítulo 1: Fundamentación Teórica

---

Es por eso que existen muchas definiciones de calidad de software según diversos autores reconocidos mundialmente. Una definición bien completa y abarcadora es la brinda el Dr. Roger Pressman una autoridad reconocida internacionalmente en los temas mejoras del proceso de software y en las tecnologías de la ingeniería de software cuando expresa “*la calidad de software* es la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente ” **(PRESSMAN, 2005)**.

La calidad del software se puede definir como la capacidad de lograr los objetivos del proyecto que estamos realizando. La norma ISO 8402 define la calidad como: “el conjunto de características de una entidad que le otorgan la capacidad de satisfacer necesidades expresas e implícita” **(LOVELLE, 1999)**. La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integrada. La calidad del software es medible y varía de un sistema a otro o de un programa a otro **(CARRASCO et al., 1995)**.

Según el estándar 610-1990 del Instituto de Ingenieros Eléctricos y Electrónicos(IEEE, sus siglas en inglés) la calidad del software "es el grado con el que un sistema, componente o proceso cumple los requerimientos y las necesidades o expectativas del cliente o usuario" **(GUERRA y OCA, 2005)**.

## 1.2.3 Pruebas de Software.

Las pruebas de software son procesos que permiten verificar y revelar la calidad de un producto software. Las mismas se integran dentro de las diferentes fases del ciclo del software. Mediante técnicas experimentales se trata de descubrir que errores tiene. Para determinar el nivel de calidad se deben efectuar unas pruebas que permitan comprobar el grado de cumplimiento respecto de las especificaciones iniciales del sistema.

Las pruebas de software son un proceso usado para identificar posibles fallos de implementación, calidad, o usabilidad de un programa. Según expresa Dijkstra "Las pruebas pueden probar la presencia de errores pero no la ausencia de ellos" **(AGUILAR, 2008)**.

Hay muchos planteamientos a la hora de abordar las pruebas de software. Una definición de pruebas de software podría ser: “...proceso de evaluación de un producto desde un punto de vista crítico donde el probador somete al producto a una serie de acciones inquisitivas, y el producto responde con su

# Capítulo 1: Fundamentación Teórica

---

comportamiento como reacción ..... además probar es buscarle los fallos a un programa” (MONTROYA, 2008).

En el sitio e-Quality Corp. consideran que las pruebas de software son: “el proceso que corre en paralelo al proceso de desarrollo de software y que se realiza por el convencimiento de que todo sistema debe ser revisado con el objetivo de establecer el nivel de la calidad requerido”. Este proceso debe llevarse a cabo de una manera sistemática, y debe respaldarse en métricas bien definidas (SOFTWARE, 2007).

Según expresa María Clara Choucair Cárdenas en su artículo las pruebas de software son : “proceso realizado concurrentemente a través de las diferentes etapas de desarrollo de software que utiliza cuyo objetivo es apoyar la disminución del riesgo de aparición de fallas y faltas en operación” (CÁRDENAS, 2007) .

Las pruebas de software deben comenzar en la fase misma de la definición de requerimientos y terminar con la finalización de la aplicación de los casos de prueba al sistema ejecutable. Esto es, la prueba de software representa un subproceso que permea prácticamente todo el proceso de desarrollo de software (SOFTWARE, 2007):

- **Fase de Requerimientos:** se verifica si se siguen estándares, y si los requerimientos son completos, consistentes, rastreables, probables y modulares.
- **Fase de Diseño:** verificar si las propuestas de diseño satisfacen los requerimientos, si presentan características deseables, y si se siguieron los estándares de diseño. En particular, verificar si la arquitectura del sistema y de la base de datos son apropiadas, si los diseños tienen alta cohesión y bajo acoplamiento, y si la base de datos tiene la normalización adecuada (no redundancia ni problemas de integridad referencial).
- **Fase de Construcción (programación):** verificar si se siguen estándares, si los programas son consistentes con el diseño, y si el código es modular, además de cohesivo y bajo en acoplamiento; eventualmente, verificar el nivel y calidad de rehúso.

Las pruebas de software generan un gran cúmulo de información muy valiosa acerca de la madurez del producto de software que permite tomar decisiones importantes y dan una panorámica de la calidad y el nivel de satisfacción del mismo.

## 1.2.4 Proceso de desarrollo de software.

Un proceso de desarrollo de software tiene como propósito la producción eficaz y eficiente de un producto software que reúna los requisitos del cliente. Dicho proceso, en términos globales se muestra en la Figura 1 (**JACOBSON et al., 2000**). Este proceso es intensamente intelectual, afectado por la creatividad y juicio de las personas involucradas (**SOMERVILLE, 2002**).

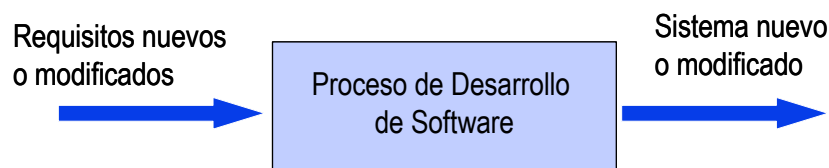


Figura 1 Proceso de desarrollo de Software (tomado del libro “El proceso Unificado de Desarrollo de Software”).

Un proceso de software es un conjunto de actividades llevadas a cabo para gestionar, desarrollar y mantener sistemas de software. (Acuña 2005).

En el libro “El Proceso Unificado de Desarrollo de Software”, se expresa que el proceso de desarrollo de software en particular, se puede definir como “el conjunto completo de actividades necesarias para convertir los requisitos de usuario en un conjunto consistente de artefactos que conforman un producto software, y para convertir los cambios sobre esos requisitos en un nuevo conjunto consistente de artefactos” (**JACOBSON et al., 2000**).

El Dr. Roger Pressman caracteriza un proceso de desarrollo de software como se muestra en la Figura 2. Los elementos involucrados se describen a continuación (**PRESSMAN, 2005**):

- **Un marco común del proceso**, definiendo un pequeño número de actividades del marco de trabajo que son aplicables a todos los proyectos de software, con independencia del tamaño o complejidad.
- **Un conjunto de tareas**, cada uno es una colección de tareas de ingeniería del software, hitos de proyectos, entregas y productos de trabajo del software, y puntos de garantía de calidad, que permiten que las actividades del marco de trabajo se adapten a las características del proyecto de software y los requisitos del equipo del proyecto.

# Capítulo 1: Fundamentación Teórica

- **Las actividades de protección**, tales como garantía de calidad del software, gestión de configuración del software y medición, abarcan el modelo del proceso. Las actividades de protección son independientes de cualquier actividad del marco de trabajo y aparecen durante todo el proceso.

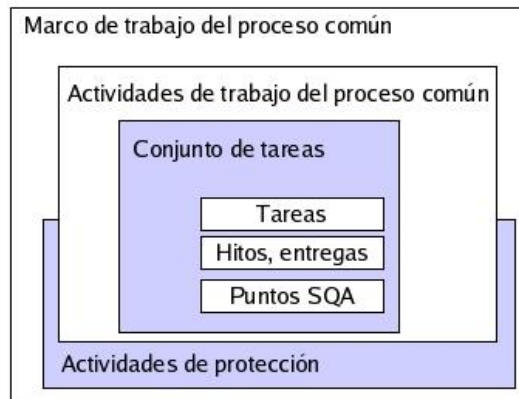


Figura 2 Elementos del proceso de desarrollo de software (tomado del libro "Ingeniería de Software .Un enfoque práctico")

El proceso de desarrollo de software no es único. No existe un proceso de software universal que sea efectivo para todos los contextos de proyectos de desarrollo. Debido a esta diversidad, es difícil automatizar todo un proceso de desarrollo de software.

A pesar de la variedad de propuestas de proceso de software, existe un conjunto de actividades fundamentales que se encuentran presentes en todos ellos (**WESLEY y SOMERVILLE, 2002**).

- **Especificación de software:** Se debe definir la funcionalidad y restricciones operacionales que debe cumplir el software.
- **Diseño e Implementación:** Se diseña y construye el software de acuerdo a la especificación.
- **Validación:** El software debe validarse, para asegurar que cumpla con lo que quiere el cliente.
- **Evolución:** El software debe evolucionar, para adaptarse a las necesidades del cliente.

Además de estas actividades fundamentales, el Dr. Roger S. Pressman menciona un conjunto de actividades protectoras, que se aplican a lo largo de todo el proceso del software. Ellas se señalan a continuación (**PRESSMAN, 2005**):

- Seguimiento y control de proyecto de software.

- Revisiones técnicas formales.
- Garantía de calidad del software.
- Gestión de configuración del software.
- Preparación y producción de documentos.
- Gestión de reutilización.
- Mediciones.
- Gestión de riesgos.

Se puede decir que el proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requerimientos del usuario en un sistema informático. Todos los conceptos anteriormente planteados de una forma u otra ayudan a comprender más sobre el proceso de desarrollo de software.

### 1.2.5 Modelos de procesos en el desarrollo de software.

Un modelo de procesos es un conjunto estructurado de elementos que describen características de procesos efectivos y de calidad. Un modelo indica “Qué hacer”, no “Cómo hacer”, ni “Quién lo hace”. Un modelo proporciona **(HUACOTO, 2005)**:

- Un punto donde comenzar.
- El beneficio de las experiencias pasadas de la comunidad participante.
- Un lenguaje común y una visión compartida.
- Un marco para priorizar acciones.
- Una forma de definir lo que significa “mejora” para la organización.

Para cada tipo de empresa, ya sea de materias, de servicio o de productos intangibles se hace muy necesario tener un modelo de proceso establecido que ayude a controlar la calidad del producto que se pone a consideración del usuario o cliente.

Generalmente en las empresas o personas que desarrollan software de cualquier tipo no cumplen los modelos de procesos, para la realización del software, algunas veces estos modelos son

desconocidos. Por este motivo se incurren en errores de diferentes tipos dando lugar al incremento de los costos, lo cual conlleva a una vida muy corta del software.

Un modelo de procesos se usa para establecer y priorizar objetivos de mejora, mejorar los procesos y proporcionar una guía que asegure el establecimiento de procesos estables, capaces y maduros.

## 1.2.6 Mejora de proceso en el desarrollo de software.

La Mejora de Procesos de Software (SPI, sus siglas en inglés) proporciona a las organizaciones las pautas de actuación necesarias para obtener mejoras observables en su proceso de desarrollo, de manera que desarrollen productos sin defectos respetando requisitos, fechas y costes.

El Coordinador de la Sociedad de Gerencia de Ingeniería del Instituto de Ingenieros Eléctricos y Electrónicos(IEEE, sus siglas en inglés) en la Región Oriental de Colombia el señor Pablo Fernando Sánchez plantea que la mejora de procesos en el desarrollo de software es: "metodología deliberada y planeada que sigue prácticas de documentación y estandarización para capturar en papel (y en la práctica ) cada elemento del proceso de software y determinar su valor agregado" (**SÁNCHEZ, 2007**).

La mejora de proceso generalmente se basa en modelos existentes, que contienen prácticas recomendadas y sirven como guía para la mejora. Su principal objetivo es aumentar la Madurez de los procesos (grado en que están definidos, administrados, medidos y aplicados) y además aumentar la capacidad de los procesos que representa los resultados esperados que pueden ser obtenidos a partir de aplicar el proceso (**MÓNOCA y VITALI, 2005**).

Se puede resumir que la mejora de los procesos de desarrollo es una abstracción o representación (textual, gráfica, formal) en la que se capturan los aspectos más importantes de un proceso de software. Es aplicable a un proyecto particular o a una familia de proyectos. La Mejora de Procesos de Software significa optimizar la efectividad y la eficiencia mejorando también los controles, reforzando los mecanismos internos para responder a las contingencias y las demandas de nuevos y futuros clientes. Es un reto para toda empresa de estructura tradicional y para sistemas jerárquicos convencionales.

## 1.2.7 Modelo de referencia para el proceso de pruebas.

Un modelo de referencia para el proceso de pruebas define el marco de referencia necesario para poder determinar las fortalezas y debilidades del proceso de pruebas implementado por la organización (**SANZ, 2008**).

CMMi es uno de los modelos de referencia más usados en la industria del software, esto trae como consecuencia que si queremos que la mejora del proceso de prueba se produzca de forma paralela al resto de los procesos de la organización será necesario que el modelo de referencia que se escoja sea compatible e integrable con el CMMi.

Dentro de los modelos de referencia más usados y que están relacionados con la mejora del proceso de pruebas actualmente tenemos los siguientes: TMM, TMMi, TPI y TMap.

## 1.3. Principales modelos usados en el proceso de desarrollo de software.

La mayoría de las empresas de software hoy en día no establecen normas, ni objetivos, ni metas, ni políticas adecuadas para lograr que los productos elaborados cumplan primero las normas internacionales para llevar a cabo el mantenimiento o para la exportación. Que la organización mejore incrementalmente su conocimiento y el dominio completo, entre los desarrolladores, sobre el proceso de software; tercero, la tecnología utilizada sea explotada al máximo; y cuarto, se tenga un grupo de personas que continuamente mejoren el proceso y velen por la calidad en la vida del proyecto (**ZAMURIANO, 2000**).

Por estas razones, han surgido modelos del proceso de software los cuales se utilizan en el desarrollo de proyectos de software. Cada uno de estos establece la madurez por niveles que van incrementándose a medida que la empresa va cumpliendo los requisitos de entrega en tiempo de los proyectos, mejore la capacidad de administración, alcance un conocimiento continuo entre los desarrolladores actuales y nuevos sobre el proceso de software, se garantice que las actividades internas correspondan a procesos planificados, sean usables y consistentes las actividades, los procesos se actualicen y mejoren, existan pruebas pilotos, se realice un análisis de costo y beneficio para el cliente, y los roles y responsabilidades estén claros en los procesos y en la organización.

### 1.3.1 Características del Modelo de Capacidad de Madurez (Capability Maturity Model – CMM).

Durante el año 1984, el Departamento de Defensa de los Estados Unidos comienza a financiar al Instituto de Ingenieros de Software (SEI ,sus siglas en inglés ) de la Universidad Carnige Mellon



# Capítulo 1: Fundamentación Teórica

---

como Centro de Investigación y Desarrollo con el objetivo que lidere los avances para la mejora de calidad de los sistemas dependientes del software. A partir de noviembre de 1986 el SEI desarrolló una primera definición de un modelo de madurez de procesos en el desarrollo de software, que se publicó en septiembre de 1987. El mismo tiene como resultado el desarrollo y la publicación en Agosto de 1991 del Modelo de Capacidad de Madurez para Software (SW-CMM 1.0) y su última versión (v1.1) se publicó en febrero de 1993.

CMM persigue la evaluación de los procesos de desarrollo de software dentro de una organización proponiendo un plan de mejoramiento en base a una serie de niveles que van desde un proceso inmaduro hasta un proceso disciplinado, ordenado y de mejoramiento continuo.

La madurez de un proceso de software, determina en qué grado un proceso de software es explícitamente definido, administrado, medido, controlado y hecho efectivo. La madurez es un indicador de la capacidad del proceso de software para lograr sus objetivos y resultados esperados **(ROMERO, 2001)**.

CMM está conformado por niveles de madurez, que indican la capacidad del proceso. Los niveles, contienen áreas claves de proceso (KPA, sus siglas en inglés), con la ayuda de ellas se alcanza los objetivos. Las áreas están organizadas con características comunes, que se aplican a la implementación o institucionalización. Las características comunes entre las áreas contienen prácticas claves, que describen la infraestructura o las actividades **(RUBIO, 2002)**.

El CMM tiene 5 niveles de madurez:

**Nivel 1 (Inicial):** No hay proceso definido para el proceso de Software.

**Nivel 2 (Repetible):** Gestión del proceso de software (coste, planificación y funcionalidad) sigue un conjunto de áreas:

- Administración de Requisitos.
- Planificación de Proyectos de Software.
- Seguimiento y supervisión de Proyectos de Software.
- Administración de Subcontratos de Software.

# Capítulo 1: Fundamentación Teórica

---

- Aseguramiento de Calidad de Software (SQA).
- Gestión de Configuración del Software.

**Nivel 3 (Definido):** Desarrollo y Mantenimiento documentado y Estandarizado. Definido, los procesos son estándares en toda la empresa y se valida si éstos son los adecuados. El conjunto de áreas es el siguiente:

- Enfoque del Proceso de la Organización
- Definición del Proceso de la Organización
- Programa de Formación
- Gestión de Integración del Software
- Ingeniería del Producto de Software
- Coordinación entre grupos
- Revisiones periódicas.

**Nivel 4 (Gestionado):** Medidas del Producto y del Proceso. Registro de valores de Calidad. Administración, se establecen métricas de cada una de las partes del proceso y se analizan sus resultados. Las áreas son las siguientes:

- Gestión cuantitativa del Proceso.
- Gestión de Calidad del Software.

**Nivel 5 (Optimizado):** Resultados cuantificados, con opción de mejora Los procesos se optimizan continuamente y se definen estándares. Las áreas son las siguientes:

- Prevención de Defectos.
- Gestión de la Tecnología.
- Gestión de Cambios en el Proceso.

El modelo CMM ofrece una estrategia para mejorar el proceso, junto con un marco para evaluar empresas. Mejorar el proceso es establecer un rango de calidad, es una medida para otros de cómo es llevado el proceso de Software, de esta forma, durante el avance entre los niveles se va robusteciendo la fabricación de software.

### 1.3.2 Características del Modelo de Capacidad de Madurez Integrado (Capability Maturity Model Integration-CMMi).

En Diciembre del 2001, el Instituto de Ingeniería de Software (SEI) lanzo la versión 1.1 del CMMI es una nueva versión de CMM. CMMI es un modelo de Procesos, o una colección estructurada de componentes que describen las características de los procesos efectivos de software que han sido probados por la experiencia.

También, está dividido por Áreas de Proceso como CMM, cada área está compuesta por metas, cada una de estas metas está conformada por prácticas. Las metas y las prácticas son específicas o genéricas. Las prácticas corresponden a una sola Área de Proceso, mientras que las áreas específicas van a través de todas las áreas de proceso.

Las metas específicas se aplican a solo un área de proceso, mientras que las metas genéricas se aplican a más de un área de proceso. Una práctica específica es una actividad considerada importante para alcanzar una meta específica. Las prácticas genéricas son prácticas que se aplican a cualquier área de proceso, y que pueden mejorar el desempeño y control de cualquier proceso.

CMMi enseña el camino para alcanzar un nivel de madurez de la organización o un nivel de capacidad de un área de proceso. Dice que hay que hacer pero no como hacerlo. El modelo está conceptualmente diseñado para permitir, en el futuro, la integración de otras disciplinas. Sigue la idea de los niveles de madurez y hereda todas las ventajas de su precursor. La diferencia radica en que integra además de la disciplina de Ingeniería de Software, las relativas a Ingeniería de Sistemas, Desarrollo Integrado de Productos, Proceso y gestión de proveedores.

Según profesores de universidades españolas “CMMi constituye un marco de referencia de la capacidad de las organizaciones de desarrollo de software en el desempeño de sus diferentes procesos, proporcionando una base para la evaluación de la madurez de las mismas y una guía para implementar una estrategia para la mejora continua de los mismos” (VILLA *et al.*, 2004).

Entre las fortalezas que presenta la utilización de CMMi se puede destacar (MUTAFELIJA, 2003):

# Capítulo 1: Fundamentación Teórica

---

- Inclusión de las prácticas de institucionalización, que permiten asegurar que los procesos asociados con cada área de proceso serán efectivos, repetibles y duraderos.
- Guía paso a paso para la mejora, a través de niveles de madurez y capacidad.
- Transición del aprendizaje individual al aprendizaje de la organización por mejora continua, lecciones aprendidas y uso de bibliotecas y bases de datos de proyectos mejorados.

Los niveles de CMMi al igual que CMM tienen 5 niveles de

**Nivel 1 (Inicial):** Este es el nivel en donde están todas las empresas que no tienen procesos. Los presupuestos se disparan, no es posible entregar el proyecto en fechas, se tienen que quedar durante noches y fines de semana para terminar un proyecto. No hay control sobre el estado del proyecto, el desarrollo del proyecto es completamente insignificante, no saben lo que pasa en él.

**Nivel 2 (Administrado):** Quiere decir que el éxito de los resultados obtenidos se pueden repetir. La principal diferencia entre este nivel y el anterior es que el proyecto es gestionado y controlado durante el desarrollo del mismo. El desarrollo no es opaco y se puede saber el estado del proyecto en todo momento.

Los procesos que hay que implantar para alcanzar este nivel son (WESLEY y SOMERVILLE, 2002):

- Administración de Requisitos.
- Planificación de Proyecto.
- Monitoreo y Control del Proyecto.
- Administración del Acuerdo del Proveedor.
- Medición y Análisis.
- Aseguramiento de Calidad del Proceso y el Producto.
- Administración de Configuración.

**Nivel 3 (Definido):** Este nivel significa que la forma de desarrollar proyectos (gestión e ingeniería) está definida, por definida quiere decir que está establecida, documentada y que existen métricas (obtención de datos objetivos) para la consecución de objetivos concretos.

# Capítulo 1: Fundamentación Teórica

---

Los procesos que hay que implantar para alcanzar este nivel son:

- Desarrollo de requisitos.
- Solución Técnica.
- Integración del producto.
- Verificación.
- Validación.
- Desarrollo y mejora de los procesos de la organización.
- Definición de los procesos de la organización.
- Planificación de la formación.
- Gestión de riesgos.
- Análisis y resolución de toma de decisiones.

La mayoría de las empresas que llegan al nivel 3 paran aquí, ya que es un nivel que proporciona muchos beneficios y no ven la necesidad de ir más allá porque tienen cubiertas la mayoría de sus necesidades.

**Nivel 4 (Administrado Cuantitativamente):** Los proyectos usan objetivos medibles para alcanzar las necesidades de los clientes y la organización. Se usan métricas para gestionar la organización.

Los procesos que hay que implantar para alcanzar este nivel son:

- Gestión cuantitativa de proyectos.
- Mejora de los procesos de la organización.

**Nivel 5 (Optimizado):** Los procesos de los proyectos y de la organización están orientados a la mejora de las actividades. Mejoras incrementales e innovadoras de los procesos que mediante métricas son identificadas, evaluadas y puestas en práctica.

Los procesos que hay que implantar para alcanzar este nivel son:

- Innovación organizacional.
- Análisis y resolución de las causas.

# Capítulo 1: Fundamentación Teórica

Según estudios realizados por Ana Sanz, Javier García, Javier Saldaña y Domingo Gaitero en el trabajo TestPAI: Un área de proceso integrada con CMMi refleja que CMMi no proporciona el soporte necesario para la mejora de los procesos de pruebas. En relación al proceso de pruebas CMMi define 3 áreas de procesos: Integración de Producto, Validación y Verificación; sin embargo no cubren todas las necesidades del proceso de pruebas. El grado de abstracción con respecto a este proceso es muy elevado, lo trata como algo genérico mientras que necesita de una atención más centrada y una definición exhaustiva (SANZ, 2008).

## 1.4 Proceso Unificado de Desarrollo de Software.

Es la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Tiene tres características fundamentales: Define el proceso como iterativo e incremental, dirigido por casos de uso y centrado en la arquitectura. RUP ha agrupado las actividades en grupos lógicos definiendo 9 flujos de trabajo principales y en su ciclo de vida cuenta con cuatro fases: inicio, elaboración, construcción y transición, que guían cada uno de los flujos, como muestra la Figura 3.

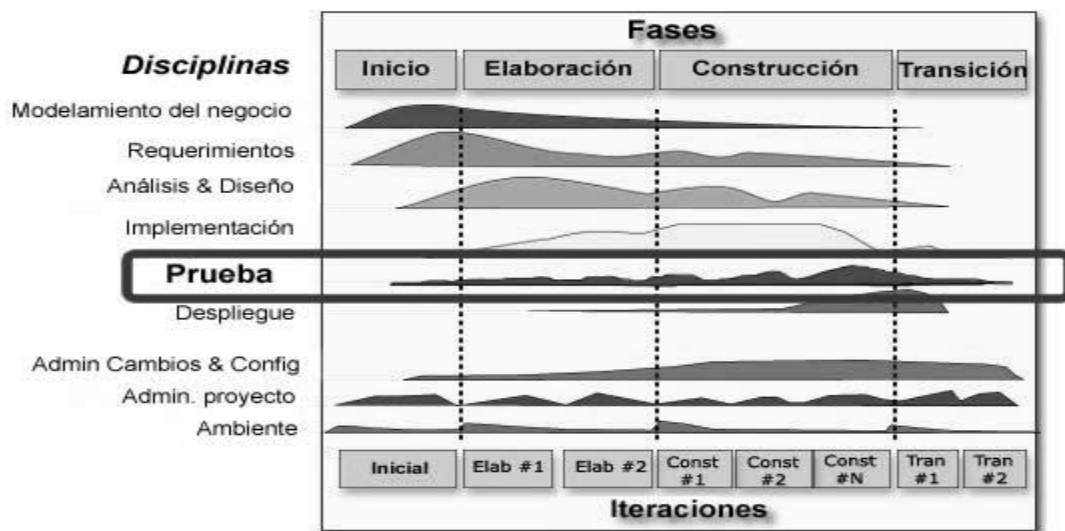


Figura 3 Ciclo de Vida de RUP (tomado del libro “El proceso Unificado de Desarrollo de Software”).

“La etapa de prueba es el flujo de trabajo fundamental cuyo propósito general es comprobar el resultado de la implementación mediante las pruebas de cada construcción, incluyendo tanto construcciones internas como intermedias, así como las versiones finales del sistema que van a ser

# Capítulo 1: Fundamentación Teórica

---

entregadas a terceras partes” (JACOBSON *et al.*, 2000). Es aconsejable que las pruebas se realicen desde el mismo momento en que comienza el desarrollo y continúe hasta que finaliza. La metodología RUP propone diferentes roles que tienen como propósito el desarrollo de las pruebas de software los mismos se relacionan a continuación:

- **Administrador de pruebas:** Es el encargado del éxito y la planificación de las pruebas.
- **Analista de pruebas:** Es responsable de identificar y definir las pruebas, así como supervisar el progreso y los resultados de las mismas.
- **Diseñador de pruebas:** Es responsable de identificar las técnicas y herramientas necesarias, así como definir y estructurar los elementos de prueba.
- **Probador:** Es el encargado de ejecutar las pruebas, analizando los resultados obtenidos.

El Proceso Unificado de Desarrollo de Software plantea 7 artefactos fundamentales para el flujo de trabajo de prueba los cuales deben ser generados dentro de esta fase, estos son:

- Modelo de Prueba
- Caso de Prueba.
- Componente de Prueba.
- Procedimiento de Prueba.
- Plan de Prueba.
- Defecto de Prueba.
- Evaluación de la Prueba.

En RUP se definen diferentes niveles de prueba, partiendo desde el nivel unidad del programa con la realización de pruebas unitarias, se realizan pruebas de integración luego de integrar cada unidad probada, después se realiza la ejecución de pruebas de sistema (dentro de la cual se incluyen las pruebas de funcionalidad, stress, usabilidad, seguridad, configuración, instalación, recuperación, documentación, entre otras) y por último se llevan a cabo las pruebas de aceptación, en las que participan los usuarios finales.

Las pruebas están presentes en todo el ciclo de vida de RUP. Durante la fase de Inicio, se pueden planificar inicialmente las pruebas a partir de la Especificación de Casos de Uso del Sistema.

En la fase de Elaboración, el análisis y diseño del software posibilitan la determinación, de forma más específica, de las pruebas a realizar. En esta fase, una vez que se hayan implementado las primeras funcionalidades, se les realizan pruebas unitarias y funcionales.

En la fase de Construcción, se concluye la implementación de todas las funcionalidades, obteniéndose un producto el cual es sometido a pruebas de integración y de sistema.

En la fase de Transición se realizan las pruebas de aceptación en las que juegan un papel decisivo los usuarios finales. En este momento es donde se decide si se acepta o no la solución de software propuesta, es decir, si responde a los requisitos establecidos por el cliente.

## 1.5 Objeto de Estudio

### 1.5.1 Descripción General del Objeto de estudio.

El proceso de desarrollo de software constituye la base principal para todo proyecto independientemente del tipo que sea. El proceso de desarrollo define quien está haciendo QUÉ, CUÁNDO y CÓMO alcanzar un determinado objetivo que es un producto que cumpla con las expectativas que espera el cliente.

El proceso de desarrollo de software en nuestra facultad es se basa en utilización de la metodología RUP un gran paso de avance ya que esta la misma contribuye a mejorar la productividad del equipo de trabajo, definiendo claramente sus actividades, roles y responsabilidades, desde los jefes de proyectos a los analistas y desde los desarrolladores a los probadores. RUP propone un conjunto de mejores prácticas a todos los miembros del equipo de proyecto como son: desarrollo de software iterativo e incremental, utilización de arquitectura basada en componentes (diseño de arquitectura flexible), modelamiento de requisitos y diseño de software (con la utilización de UML), y verificación de la calidad.

En los proyectos de la facultad se tienen en cuenta todas las fases de construcción de un software, pero hay que señalar que no se tiene en cuenta el proceso de pruebas que prácticamente pasa desapercibido. El mismo es considerado como una continuación de la fase de implementación, un error que se comente en todos los proyectos de la facultad. Esto trae como consecuencia que los proyectos al final en la fase de revisión por el grupo de calidad se encuentren errores que podían ser arreglados antes de liberar el producto y que a veces estos errores son muy costos ya que demoran la entrega a



tiempo del producto, obligando a los desarrolladores o al proyecto en general trabajar en jornadas muy agotadoras.

## 1.5.2 Situación Problemática.

La Universidad de las Ciencias Informáticas (UCI, sus siglas) actualmente está tratando de implantar en el proceso de desarrollo de software el nivel 2 de CMMi y CMMi no satisface la calidad del proceso de pruebas. Nuestra facultad no está ajena a eso por lo que su principal meta es implantar un modelo de calidad que le de soporte al proceso de pruebas, ya que las mismas constituyen un pilar fundamental en la calidad del producto desarrollado.

La disciplina de pruebas es considerada actualmente en los proyectos productivos de la Facultad 9 como una continuación de la implementación del sistema, no existiendo para la misma una estrategia trazada ya sea a nivel de proyecto o a nivel de organización, facultad en nuestro caso, que permita desarrollar un seguimiento o mejora de los resultados obtenidos y comparaciones de las métricas obtenidas con otros proyectos similares.

Aún cuando se hace uso de un plan de pruebas que sugiere la metodología RUP, lo cual es considerado como una buena práctica, el mismo se completa sobre la base de la necesidad de entregar una documentación para las auditorías, lo que impide la realización correcta y suficiente de un cronograma de pruebas dentro de los proyectos, lo que se traduce en señalamientos durante el proceso de revisiones por entes externos a los proyectos. A esto se le añade que no se trabaja sobre un enfoque a las pruebas durante las diferentes etapas del proyecto, lo cual impide la identificación a tiempo de diferentes errores tanto de código, como de gestión, o en la documentación mantenida del proyecto.

Aún cuando los planes de prueba se puedan cumplir tal y como están escritos se puede encontrar casos donde el producto final no cumpla con la calidad requerida ya sea por incumplimiento de los requisitos funcionales o requisitos no funcionales o los errores en la documentación. A esto se le ha de sumar que luego sobre la base de estos aspectos identificados la facultad no se propone ninguna mejora de los procesos de pruebas pues no los tiene definidos ni controlados y por lo tanto es incapaz de medirlos.

La facultad no cuenta con un plan de pruebas o una estrategia que permita desarrollar un control a tiempo completo del estado y completitud de los proyectos, validando mediante buenas prácticas y objetivos cumplidos la calidad de los productos que se desarrollan.

### 1.5.3 Análisis de otras soluciones existentes.

Dentro de las principales soluciones existentes que brinda parcialmente o totalmente respuesta al problema planteado se encuentran.

1. El Trabajo realizado por Ana Sanz, Javier Saldaña, Javier García todos del pertenecientes al Departamento de Informática de la Universidad Carlos III de Madrid España y el profesor Domingo Gaitero de ATOS.

El título del trabajo es TestPAI: Un área de proceso de pruebas integrada con CMMi.

El TestPAI es un área de proceso totalmente definida que puede ser integrada con CMMi para la mejorar el proceso de pruebas de forma paralela a la mejora de los otros procesos de software. TestPAI tiene una estructura similar a CMMi e incluye todas las prácticas con las pruebas. Además TestPAI ha sido implementada satisfactoriamente en una organización real.

2. El trabajo de diploma para optar por título profesional de ingeniero en sistemas realizado por Daniel Rolando Valdivia Espinoza y Eduardo Geonias Valdivia Espinoza ambos de la Facultad de Ingeniería de Sistemas e Informática de la Universidad Nacional mayor de San Marcos Lima Perú.

El título del trabajo es Estándares de calidad para prueba de software.

El trabajo es propuesta de un modelo de procesos de pruebas con el objetivo de alcanzar el Nivel 2 del Modelo de Madurez de Prueba (TMM, sus siglas en inglés). Incluye la mejores prácticas de pruebas y el aseguramiento de la calidad de software.

### 1.6 Conclusiones Parciales.

En este capítulo se da toda la fundamentación teórica que sustenta esta investigación, se hizo un estudio de las principales definiciones que se encuentran asociadas al dominio del problema, se analizaron los principales modelos de calidad más usados actualmente. Donde se analizó que modelos como CMMi no le dan soporte al proceso de pruebas lo que nos ve en la necesidad proponer un modelo de calidad que a la vez se integre con el mismo y mejore el proceso de prueba.

Además se hace un análisis detallado de la situación problemática donde se conoce la verdadera situación que tiene la facultad en cuanto al tema de las pruebas. Se realiza una descripción detallada del objeto de estudio donde se conoció como la facultad realiza el proceso de desarrollo de software.

# *Capítulo 1: Fundamentación Teórica*

---

Finalmente se hizo un estudio de otras soluciones existentes para saber y tener conocimientos de otras propuestas que han realizado diferentes personas referentes al tema tratado en esta investigación.

## Capítulo 2

### Capítulo 2: Solución Propuesta.

#### 2.1 Introducción.

En este capítulo se describen las características fundamentales del *framework* Modelo Integrado de Madurez de Prueba (TMMi, sus siglas en inglés) que constituye la solución para esta investigación. Se realiza una caracterización de las áreas de proceso para el nivel 2 de TMMi, se define una estrategia para adoptar el proyecto de mejora en la facultad y se propone un curso de capacitación.

#### 2.2 Caracterización del Modelo Integrado de Madurez de Prueba.

##### 2.2.1 Antecedentes e Historia.

El *framework* TMMi ha sido desarrollado por la Fundación TMMi como guía y referencia para la mejora del proceso de prueba y se posiciona como un complemento al modelo CMMI (VEENENDAAL, ERIK VAN, 2008).

Al igual que el CMMI, el TMMi también usa el concepto de niveles de madurez para la evaluación y mejora del proceso de prueba. Son identificados claramente las áreas de proceso, las metas y las prácticas tanto genéricas como específicas. Con la aplicación de los criterios de madurez TMMi mejorará el proceso de prueba, lo que potenciará un impacto positivo a la calidad de producto. El TMMi ha sido desarrollado para apoyar organizaciones en evaluación y mejoramiento de su proceso de prueba. Dentro del TMMi, las pruebas evolucionan de un proceso caótico, indefinido con una carencia de recursos, herramientas y probadores instruidos a un proceso maduro y controlado que tiene la prevención de defecto como su objetivo principal.

El TMMi proporciona un enfoque más amplio y estructurado al proceso de prueba, existe un gran número de experiencias positivas con el modelo TMMi y las organizaciones que la utilizan van en aumento, las experiencias muestran que aplicar el modelo de madurez TMMi tiene un impacto positivo sobre la calidad del producto.

De manera similar al CMMI el TMMI nos presenta una mejora en los procesos relacionados con la calidad del software, definiendo 5 niveles de madurez. Cada nivel de madurez contiene un conjunto de áreas de proceso bien definidas. Estos niveles definen la jerarquía de madurez y el camino de evolución a la mejora del proceso de prueba.

### 2.2.2 Fuentes.

Para el desarrollo de TMMi se ha utilizado el framework Modelo de Madurez de Prueba (TMM, sus siglas en inglés) desarrollado por el Instituto de Tecnología de Illinois. TMMi se guía en gran medida por la labor realizada en el Modelo Integrado de Capacidad de Madurez.

El TMMi ha sido desarrollado como un modelo por etapas o niveles y el mismo establece áreas de proceso para definir un camino para mejorar una organización. Este camino de mejora está descrito por un componente llamado nivel de madurez. Un nivel de madurez bien definido es un buen comienzo hacia el logro de las mejoras de procesos de una organización.

Otras fuentes en el desarrollo TMMi se pueden incluir:

- La evolución del Modelo de Prueba Gelperin Hetzel (que describe la evolución del proceso de prueba durante un periodo de 40 años **(GELPERIN y HETZEL, 1998)** ).
- El modelo de prueba Beizer, que describe la evolución de la persona probadora del pensamiento **(BEIZER, 1990)**.
- La investigación sobre TMM realizada por el MB-TMM financiado por Unión Europea.
- Estándar IEEE 829 para la documentación de pruebas de software **(IEEE\_829, 1998)**.
- La Mejora del Proceso de Prueba de(TPI, sus siglas en inglés) Tim Koomen y Martin Pol **(KOOMEN y POL, 1999)** ).

### 2.2.3 Ámbito de aplicación de TMMi.

El framework TMMI tiene un intenso campo de aplicación, en los cuales el mismo aporta una amplia gama de experiencias positivas. Los mismo se describen a continuación:

### 2.2.3.1 Ingeniería de Sistemas y de Software.

El TMMi es destinado para apoyar actividades de pruebas y mejora de proceso de prueba tanto de la ingeniería de sistemas como de disciplina de ingeniería de software. La ingeniería de sistemas cubre el desarrollo de sistemas totales, que pueden o pueden no incluir el software. La ingeniería de software cubre el desarrollo de sistemas de software.

### 2.2.3.2 Nivel de Prueba.

Mientras que algunos modelos usados para la mejora del proceso de prueba tratan la mejora de la prueba principalmente en las pruebas de alto nivel como por ejemplo el modelo Mejora de Proceso de Prueba (TPI, sus siglas en inglés) (**KOOMEN y POL, 1999**) o un aspecto de la estructuras de prueba por ejemplo la organización de prueba, el TMMi se dirige a todos los niveles de prueba (incluso pruebas estáticas) y aspectos de pruebas estructuradas. Con el respeto a pruebas dinámicas, tanto las pruebas de bajo nivel como las pruebas de alto nivel están dentro del ámbito de aplicación del TMMi. TMMi se dirige a cuatro piedras angulares para pruebas estructuradas (el ciclo de vida, las técnicas, la infraestructura, y la organización) (**VEENENDAAL, ERIK VAN y TEUNISSEN, 2002**).

### 2.2.3. 3 TMMi y CMMi.

También es importante señalar que TMMi se posiciona como un modelo complementario al CMMI. En muchos casos un nivel TMMi dado necesita el apoyo específico de áreas de proceso a su nivel CMMI correspondiente o de un nivel más abajo de CMMI. En casos excepcionales hay hasta una relación a más alto niveles de CMMI. Las áreas de proceso y las prácticas que son elaboradas dentro del CMMI no son repetidas dentro de TMMi.

### 2.2.3.4 Evaluaciones.

Muchas organizaciones encuentran el valor en el marcaje de su progreso en la mejora de proceso de prueba tanto con objetivos internos como con clientes externos y proveedores. Las evaluaciones de proceso de prueba se concentran en identificar oportunidades de mejora y entender la posición de la organización con relación al modelo seleccionado o estándar. El TMMi provee un modelo de referencia excelente para ser usado durante tales evaluaciones. Los equipos de evaluación usan TMMi para dirigir su identificación y asignación de prioridades de conclusiones. Estas conclusiones con la dirección de prácticas TMMi son usadas para planear mejoras para la organización. El marco de evaluación en sí mismo no es la parte del TMMi. Las exigencias para evaluaciones TMMi son

descritas por la Fundación TMMi. Estas exigencias están basadas en la ISO 15504. El logro de un nivel de madurez específico debe significar la misma cosa para organizaciones diferentes. Las reglas para asegurar esta consecuencia están contenidas en las exigencias de método de evaluación TMMi. Las exigencias de método de evaluación TMMi contienen pautas para varias clases de evaluaciones, evaluaciones por ejemplo formales, las exploraciones rápidas y autovaloraciones.

### **2.2.4 Niveles de Madurez.**

TMMi tiene una estructura organizada para la mejora del proceso de prueba. Contiene las etapas o niveles por los cuales una organización va evolucionando su proceso de prueba de uno caótico y no manejado, a uno que es manejado, definido, medido, y optimizado. El logro de cada una de las etapas donde se asegura una adecuada mejora se establece como base para la siguiente fase.

La estructura interna del TMMi es rica en las pruebas que pueden ser aprendidas y aplicadas de forma sistemática para apoyar gradualmente al proceso de mejora. Existen 5 niveles que describen la jerarquía de madurez del TMMi y un camino evolutivo para poner a prueba el proceso de mejora. Cada nivel tiene un conjunto de áreas de proceso donde la organización debe centrarse para lograr la madurez en ese nivel.

Debido a que cada nivel de madurez es una base para el siguiente nivel, es generalmente erróneo si se trata de saltar algún nivel de madurez. Al mismo tiempo debemos reconocer que en la mejora de los procesos de prueba los esfuerzos deberían centrarse en las necesidades de la organización en el contexto de su entorno empresarial y en las áreas de proceso de alto nivel de madurez. Las áreas de proceso para cada nivel madurez de TMMi se muestran en la siguiente figura (ver Anexo# 4) y se describen más adelante.

Dentro de TMMi las herramientas de prueba son tratados como un recurso de apoyo (práctica) y son por lo tanto parte del área de proceso donde ellos proporcionan el apoyo. Por ejemplo aplicando una herramienta de diseño de prueba es un recurso de apoyo dentro del área de proceso Diseño de Prueba y Ejecución al nivel 2 TMMi.

#### **2.2.4.1 Nivel Inicial.**

En el nivel 1 de TMMi las pruebas son un caos y el proceso definido es considerado parte de la depuración. La organización no suele proporcionar un entorno estable para apoyar los procesos. El éxito en estas organizaciones depende de la competencia y de la entrega de las personas a la

organización y no del uso de procesos. Las pruebas se realizan después que la codificación se ha completado. Pruebas y depuración son intercalados para obtener los errores del sistema. El objetivo de las pruebas de este nivel es demostrar que el software funciona sin grandes fracasos. Los productos son puestos en libertad sin suficiente visibilidad respecto a la calidad y al riesgo. En el despliegue, el producto a veces no suele cumplir las necesidades, no es estable, o es demasiado lento para trabajar.

El nivel 1 de TMMi no tiene definido ninguna área de proceso. Las organizaciones se caracterizan por una tendencia a cometer más errores, al abandono de los procesos en un momento de crisis y la imposibilidad de repetir sus éxitos. Lo que implica que la calidad del software no tenga las expectativas esperadas por los usuarios.

### **2.2.4.2 Nivel 2 Gestionado.**

En el nivel 2 TMMi, las pruebas hacen un proceso manejado y son claramente separadas de la depuración. Se establece una estrategia de prueba para toda la organización. Se comienza a desarrollar un plan de prueba que ofrece un acercamiento a las pruebas, este acercamiento está basado en el resultado de una evaluación de riesgo de los productos. Las técnicas de dirección de riesgo son usadas para identificar los riesgos de los productos basados en exigencias documentadas. El plan de prueba define que pruebas se requieren, cuando, cómo y por quien. Se establecen los compromisos con todo el personal involucrado en la mejora del proceso. Las pruebas son supervisadas y controladas para asegurar que van según el plan y las acciones pueden ser tomadas si las desviaciones ocurren. Las pruebas todavía pueden comenzar relativamente tarde en el ciclo de vida de desarrollo, por ejemplo durante el diseño o hasta durante la fase de codificación. Se definen los niveles de prueba (unidad, la integración, el sistema y aceptación). Para cada nivel de prueba identificado hay objetivos de pruebas específicos definidos en la estrategia de prueba por toda la organización.

El objetivo principal de pruebas en unas organizaciones del nivel 2 TMMi es verificar que el producto satisface las exigencias especificadas. El objetivo también es diferenciar claramente los procesos de pruebas y depuración. Muchos problemas de calidad a este nivel de TMMi ocurren porque las pruebas ocurren tarde en el ciclo de vida de desarrollo. Los defectos son propagados de las exigencias del cliente al diseño y posteriormente al código.

Las áreas de proceso para este nivel son:



- Política de Prueba y Estrategia.
- Prueba de Planificación.
- Prueba de Seguimiento y Control.
- Diseño de Prueba y Ejecución.
- Prueba de Medio Ambiente.

### 2.2.4.3 Nivel 3 Definido.

En el nivel 3 TMMi, las pruebas ya no son una fase que sigue a la codificación. Son totalmente integradas al ciclo de vida de desarrollo de software y los roles asociados. La planificación de prueba es hecha en una etapa temprana del proyecto, por ejemplo durante la fase de requerimientos, por medio de un plan de prueba maestro.

El desarrollo de un plan de prueba maestro añade las habilidades de planificación de prueba adquiridas al nivel 2. Se utilizan por las organizaciones procesos de pruebas estándares, que son la base para el nivel 3 de madurez, que son establecidos y mejorados con el tiempo. Una organización de prueba y un programa de formación de prueba específico existen, y las pruebas son percibidas como una profesión. Los casos de prueba son juntados, almacenados y manejados en una base de datos central para pruebas de regresión y reutilización. Los métodos básicos apoyan actividades de pruebas claves. Las organizaciones a este nivel comienzan a realizar revisiones para el control de calidad; un programa de revisión formal es puesto en práctica aunque todavía no tiene unido un proceso de pruebas dinámico.

Las revisiones ocurren a través del ciclo de desarrollo del software. Los profesionales de prueba están implicados en revisiones según las especificaciones hechas por los clientes. Por lo cual los diseños de prueba del nivel 2 de TMMi que son principalmente pruebas de funcionalidad, diseños de prueba y técnicas de prueba son ampliados, dependiendo los objetivos comerciales de la organización, para incluir también pruebas no funcionales, por ejemplo en utilidad y fiabilidad.

Una distinción crítica entre el nivel 2 y 3 de madurez TMMi es el alcance de los estándares, descripciones de proceso, y procedimientos. Al nivel 2 de madurez éstos pueden ser completamente diferentes en cada caso específico. Otra distinción crítica es que en el nivel 3 de madurez, los

procesos son descritos más rigurosamente que al nivel 2 de madurez. Como una consecuencia al nivel 3 de madurez, la organización debe revisar de nuevo las áreas de proceso del nivel 2 de madurez.

Las áreas de proceso para el nivel 3 de TMMi son las siguientes:

- Prueba de Organización.
- Programa de Formación de Prueba.
- Ciclo de vida de Prueba e Integración.
- Pruebas No funcionales.
- Par de Revisiones.

#### **2.2.4.4 Nivel 4 Gestión y Medición.**

En el nivel 4 de TMMi las pruebas son un proceso bien definido y bien fundado. La organización y los proyectos establecen objetivos cuantitativos para lograr una mayor calidad de producto y establecen criterios para guiar estos objetivos. La calidad de producto y la interpretación de proceso son entendidas en términos estadísticos y son manejadas en todas partes del ciclo de vida de la mejora del proceso de pruebas. Las revisiones y las inspecciones se consideran partes de las pruebas

Las revisiones están formalmente usadas como medios para controlar puercas de calidad. Los productos son evaluados usando criterios cuantitativos para atributos de calidad, como fiabilidad de utilidad y capacidad de mantenimiento. Un amplio programa de medidas para las pruebas proporciona información y visibilidad en cuanto al proceso de prueba. Las pruebas son percibidas como la evaluación.

Las áreas de proceso para el nivel 4 de TMMi son las siguientes:

- Medida de Prueba.
- Evaluación de la Calidad del Producto.
- Par de Revisiones Avanzada.

#### **2.2.4.5 Nivel 5 Optimización.**

Sobre la base de todos los resultados alcanzados mediante el cumplimiento de todos los objetivos de mejora de los anteriores niveles de madurez, la prueba es ahora un proceso completamente definido y es capaz de controlar los costes y la eficacia de las pruebas. En nivel 5 de madurez de TMMi, una

organización mejora continuamente sus procesos sobre la base de una comprensión cuantitativa de la causa común de variación inherente en los procesos.

Los métodos y las técnicas son optimizados y hay una atención continua en la mejora de proceso de prueba. La prevención de defecto y el control de calidad son practicados. Las pruebas estadísticas, las medidas de niveles de confianza, honradez, y fiabilidad conducen el proceso de prueba. El proceso de prueba es caracterizado por probar las medidas de calidad. Un procedimiento detallado existe para seleccionar y evaluar herramientas de prueba. Las herramientas apoyan el proceso de prueba tanto como sea posible.

La reutilización de proceso también es practicada al nivel 5 apoyado por una biblioteca de activo de proceso. Las pruebas son un proceso con el objetivo prevenir defectos.

Las áreas de procesos del nivel 5 de TMMi son las siguientes:

- Prevención de Defectos.
- Optimización de Procesos de Prueba.
- Control de la Calidad.

### **2.2.5 Estructura del TMMi.**

La estructura del TMMi se basa en gran parte basada en la estructura del CMMI. Esto es una ventaja principal porque mucha gente y organizaciones ya están familiarizadas con la estructura CMMI. La estructura CMMI hace una distinción clara entre prácticas que se requieren (metas) o recomendadas (prácticas específicas, productos de trabajo típicos, etc.) poner en práctica. Este aspecto también es incluido en el TMMi.

#### **2.2.5.1 Componentes requeridos, esperados e Informativos.**

Los componentes se dividen en tres grandes grupos: requeridos, esperados e informativos.

##### **2.2.5.1.1 Componentes requeridos.**

Los componentes requeridos describen lo que una organización debe conseguir para satisfacer un área de proceso. Este logro debe ser visiblemente puesto en práctica en los procesos de una organización. Los componentes requeridos en TMMi son las metas específicas y genéricas. La satisfacción de las metas nos da a conocer si un área de proceso ha sido conseguida y satisfecha.

### **2.2.5.1.2 Componentes esperados.**

Los componentes esperados describen lo que una organización pondrá en práctica típicamente para conseguir un componente requerido. Los componentes esperados dirigen a aquellos que ponen en práctica mejoras o realizan evaluaciones. Los componentes esperados incluyen tanto prácticas específicas como genéricas.

### **2.2.5.1.3 Componentes informativos.**

Los componentes informativos proporcionan detalles que ayudan a organizaciones a empezar en el pensamiento sobre como acercarse a los componentes requeridos y esperados. Las subprácticas, los productos de trabajo típicos, las notas, los ejemplos, y las referencias son todos los componentes informativos.

### **2.2.5.2 Componentes del TMMi.**

El modelo TMMi requiere de una serie de componentes que se pueden ilustrar en la siguiente figura (ver Anexo # 5) .Los mismos se describirán en las secciones siguientes.

#### **2.2.5.2.1 Niveles de madurez.**

Un nivel de madurez dentro del TMMi puede ser considerado como el grado de la calidad del proceso de prueba. Es definido como un paso evolutivo en la mejora de proceso de prueba. Cada nivel desarrolla una parte importante del proceso de prueba de la organización. Hay cinco niveles de madurez dentro del TMMi. Cada nivel de madurez dice que poner en práctica a fin de conseguir el nivel dado. Mientras más alto sea el nivel de madurez en el cual la organización es clasificada, más maduro es el proceso de prueba de la organización. Para alcanzar un nivel de madurez particular, una organización debe satisfacer todos los objetivos apropiados (tanto específico como genérico) de las áreas de proceso al nivel específico y también aquellos a niveles de madurez más tempranos.

#### **2.2.5.2.2 Áreas de proceso.**

Cada nivel de madurez de TMMi consta de varias áreas de proceso con la excepción del nivel 1, estas áreas de procesos son donde la organización debe centrarse para mejorar los procesos de prueba. Identificar las áreas de proceso son las cuestiones donde se debe centrar la organización para lograr un nivel de madurez. Cada área de proceso identifica un grupo de pruebas. Las áreas de procesos que se identifican dentro del TMMi constituyen el principal factor para determinar la capacidad del proceso

de prueba. Todas las áreas de un determinado nivel de madurez y los niveles anteriores deben cumplirse para considerar que se ha alcanzado un determinado nivel.

### **2.2.5.2.3 Metas específicas.**

Una meta específica describe una única característica que debe presentar para satisfacer el área de proceso. Una meta específica se utiliza para determinar si un área de proceso está completamente satisfecha.

### **2.2.5.2.4 Metas genéricas.**

Las metas genéricas aparecen cerca del final de un área de proceso y son llamadas genéricas porque la misma declaración de la meta aparece en múltiples áreas de procesos. Una meta genérica describe las características que deben de estar presentes para institucionalizar los procesos que se ponen en práctica en un área de proceso. La meta genérica se usa para determinar si un área de proceso está satisfecha.

### **2.2.5.2.5 Prácticas específicas.**

Una práctica específica es la descripción de una actividad que se considera importante para lograr la meta específica asociada. La práctica específica describe las actividades esperadas que hay que realizar en el logro de las metas específicas de un área de proceso. Una práctica específica es un componente de espera.

### **2.2.5.2.6 Productos de trabajo Típicos.**

Los productos de trabajo típicos son el resultado que se obtiene al realizar las prácticas específicas. Un producto de trabajo típico es un componente informativo.

### **2.2.5.2.7 Sub-Prácticas.**

Una subpráctica es una descripción detallada que proporciona la dirección a interpretar y poner en práctica una práctica específica. Las subprácticas son realmente un componente informativo destinado sólo proporcionar ideas que pueden ser útiles para la mejora de proceso de prueba.

### **2.2.5.2.8 Prácticas genéricas.**

Las prácticas genéricas aparecen cerca del final de un área de proceso y son prácticas genéricas porque la misma práctica aparece en múltiples áreas de proceso. Una práctica genérica es la

descripción de una actividad que se considera importante para lograr la meta genérica asociada. Una práctica genérica es un componente de espera.

### 2.3 Caracterización de las Áreas de Proceso.

Las áreas de procesos a las cuales se le va a realizar la caracterización son las primeras áreas de procesos, las cuales pertenecen al nivel 2 del Modelo Integrado de Madurez de Prueba.

Lo que se pretende con el nivel 2 de TMMI es mejorar el proceso de pruebas. Las pruebas en este nivel hacen un proceso manejado y son claramente separadas de la depuración. Se define una estrategia de prueba y se desarrolla el plan de pruebas. El plan de prueba definirá que pruebas se requieren, cuando, cómo y por quien. Las pruebas son supervisadas y controladas para asegurar que van según el plan y lo que permite realizar acciones en caso de alguna desviación.

Las pruebas son niveladas: se alcanzan todos los niveles de prueba (unidad, integración, sistema y aceptación). Para cada nivel de prueba identificado hay objetivos de pruebas específicos definidos en la estrategia de prueba. El objetivo principal de pruebas en el nivel 2 TMMi es verificar que el producto satisface las exigencias especificadas. El objetivo también es diferenciar claramente los procesos de pruebas y depuración.

Se escogió el nivel 2 de TMMI debido a que el nivel 1 es en el que están todas las organizaciones, y ya en el nivel 2 una organización está caracterizada por tener un proceso de prueba, que el mismo es planificado y ejecutado según la política de prueba establecida por la organización, involucra a los principales interesados del proyecto, es monitoreado, controlado, revisado continuamente y es evaluado para su conformidad con la descripción del mismo.

De acuerdo con todo lo planteado anteriormente tenemos que comenzar por aplicar a los proyectos productivos de la facultad las áreas de procesos que establece el nivel 2 y cumplir con todas las prácticas que comprende el mismo. Ya que es erróneo saltar de un nivel a otro sin lograr la madurez del nivel anterior.

Las áreas de procesos de nivel 2 son:

- Política de Prueba y Estrategia.

El propósito de Política de Prueba y Estrategia es desarrollar y establecer una política de prueba y una estrategia de prueba en la facultad donde se definirán los niveles de prueba.

- Planificación de Pruebas.

El propósito de la Planificación de Prueba es definir un acercamiento de prueba basado en los riesgos identificados y la estrategia de prueba definida, y establecer y mantener proyectos bien fundados para realizar y manejar las actividades de pruebas.

- Pruebas de Monitoreo y Control.

El propósito de las Prueba de Monitoreo y Control es proporcionar un entendimiento del progreso de las pruebas y la calidad del producto de modo que las acciones correctivas apropiadas puedan ser tomadas cuando el progreso de prueba se desvía considerablemente del plan o la calidad de producto se desvía considerablemente de expectativas.

- Diseño de Prueba y Ejecución.

El propósito del Diseño de Prueba y Ejecución es mejorar la capacidad de proceso de prueba durante diseño de prueba y ejecución estableciendo especificaciones de diseño de prueba usando técnicas de diseño de prueba, realizando un proceso de ejecución de prueba estructurado y manejando incidentes de prueba.

- Prueba de Ambiente.

El propósito de la Prueba de Ambiente es establecer y mantener un ambiente adecuado, incluso datos de prueba, en los cuales es posible ejecutar las pruebas de un modo manejable y repetible.

### **2.4 Propuesta de Prácticas y Metas específicas para alcanzar el nivel 2 de TMMI. Área de proceso Política de Prueba y Estrategia.**

Cuando se quiere mejorar el proceso de prueba hay que definir primero una política de prueba. La política de prueba define los objetivos de prueba que se va a trazar la facultad y las estrategias en cuanto a las pruebas. Esta política de prueba debe ser alineada con la política de negocio de la facultad. Una política de prueba es necesaria para alcanzar una vista común de pruebas entre todos los involucrados en la mejora del proceso de prueba. Se declaran los objetivos para la mejora del proceso de prueba. Basado en la política de prueba se define una estrategia de prueba. La estrategia de prueba cubre las exigencias de las pruebas para la facultad. La estrategia de prueba se dirige a los riesgos que existan en los proyectos de la facultad y se debe presentar un proceso para mitigar

riesgos de acuerdo con la política de pruebas que se trace. La estrategia de prueba incluirá una descripción de los niveles de prueba que deben ser aplicados, por ejemplo: unidad, integración, sistema y prueba de aceptación. La estrategia de prueba sirve como un punto inicial para las actividades de pruebas dentro de proyectos. Cuando una estrategia de prueba es definida y seguida, se lleva a cabo un proceso de prueba más eficiente.

### 2.4.1 Propósito.

El propósito de la Política de Prueba y Estrategia es desarrollar y establecer una política de prueba y una estrategia de prueba en la facultad donde se definirán los niveles de prueba.

### 2.4.2 Alcance.

El área de proceso Política de Prueba y Estrategia implica la definición y el despliegue de una política de prueba y estrategia de prueba. Dentro de la estrategia de pruebas son identificados los niveles de prueba. Para cada nivel de prueba son definidos los objetivos de las pruebas, las responsabilidades y las tareas principales.

### 2.4.3 Prácticas y metas específicas.

Para lograr satisfacer el área de proceso Política de prueba y Estrategia hay que realizar metas específicas con el fin de lograr que el área de proceso contribuya a alcanzar el nivel de madurez trazado. Dentro de las metas específicas que define esta área de proceso encontramos: Establecer una **política de prueba**, Establecer una **estrategia de prueba**.

Para **establecer una política de prueba** la facultad debe alinear esta política con el negocio de la facultad. Esta política de negocio es establecida por el decano de la facultad. Para ello se deben desarrollar diferentes prácticas específicas dentro de la cuales tenemos: Definir las metas de la prueba, Definir la Política de Prueba y Distribuir la política de Prueba entre los involucrados.

- Para **definir las metas de la prueba** se debe hacer un estudio de las necesidades comerciales de la facultad. Para ello se debe tener en cuenta la misión de las pruebas, se realiza un estudio del negocio que se lleva a cabo en la facultad y las necesidades de los clientes en cuanto a los productos que desarrolla la facultad y se estudian las políticas del negocio que se lleva a cabo en la facultad así como las principales metas en cuanto a lograr una mayor calidad del producto que se desarrolla en la facultad. Se refinan las necesidades comerciales y los objetivos si es necesario y se definen las metas de prueba a las necesidades comerciales .Las



metas de prueba a las necesidades comerciales deben incluir la validación del producto para el caso de uso, deben prevenir defectos que pueden ocurrir, verificar la conformidad de estándares externos, proporcionar la visibilidad de la calidad del producto y un corto plazo para la ejecución de la prueba.

- **Definir la Política de Prueba** esta debe estar basada en las metas de pruebas definidas. Para ello se deben tener en cuenta los puntos de vista básicos en cuanto a las pruebas y la profesión de pruebas, se debe hacer una definición de las prueba y se debe tener en cuenta el nivel de calidad para conseguir esta política. Se deben realizar pruebas para encontrar fallos dentro de la política
- **Distribuir la Política de Prueba entre los involucrados** para esto se realiza un plan de despliegue y se hace una presentación de una política de prueba a través de una reunión con todo el personal involucrado en la mejora del proceso de prueba. Se escogió esta vía de distribuir la política de prueba y no la de enviarla por medios electrónicos porque mediante la reunión la información llega de manera más clara y concisa y se evacuan todas las dudas y se analiza con todo por si hay que mejorarla.

Para **establecer una estrategia de prueba** se debe hacer un estudio de la política de prueba y meta de pruebas trazadas por la facultad ya que la política de prueba se pone en práctica con una estrategia de prueba. Esta estrategia proporciona una visión clara de cómo van a ser probados los productos en todas partes de su ciclo de desarrollo del software. la misma debe estar basada en la metodología de desarrollo que implementa la facultad (RUP). La estrategia contiene una definición de niveles de prueba y los grupos organizativos que ejecutan ellos, el acercamiento a pruebas de regresión, entrada y criterios de salida, y más. La estrategia de prueba debe ser alineada con la Política de prueba. Para la estrategia de prueba de la Facultad se propone que se utilice la que define RUP (ver anexo # 8).

Después de realizada la estrategia se debe **distribuir la estrategia de Prueba entre los involucrados** para esto se realiza un plan de despliegue y se hace una presentación de la estrategia de prueba a través de una reunión con todo el personal involucrado en la mejora del proceso de prueba. Se escogió esta vía de distribuir la estrategia de prueba y no la de enviarla por medios electrónicos porque mediante la reunión la información llega de manera más clara y concisa y se evacuan todas las dudas, pudiéndole realizarle cambios si fuese necesario. Después de aprobada por todo el personal involucrado en la mejora del proceso de pruebas la estrategia de prueba debe de ser

de estricto cumplimiento por todos los involucrados y se registran los incumplimientos en el plan de pruebas.

### **2.5 Estrategia General para aplicar el framework TMMi a los proyecto de la Facultad 9.**

El TMMi no es sólo un modelo teórico, sino un conjunto bien definido de áreas proceso y metas (específicas como genéricas). Aunque el modelo es bastante fácil de entender, su aplicación no siempre es una tarea simple. En promedio podríamos tardar 2 años para alcanzar el TMMi nivel 2

Un proyecto de TMMi cuenta de diferentes fases:

- **Iniciación** (en esta fase se determinan el alcance y los objetivos).
- **Planificación** (en esta fase define al proyecto en términos de estructura, recursos, etc.).
- **Implementación** (en esta fase se realiza el desarrollo de entregables, procedimientos y plantillas).
- **Despliegue** (publicación, despliegue de los resultados).

#### **Fase de Iniciación.**

##### **Determinar el estado actual en que se encuentra la facultad.**

- Capacitación del personal con el objetivo de que entiendan mejor el proceso de mejora.
- Conocer el nivel de madurez actual.

Se debe realizar una evaluación del estado actual en que se encuentran los proyectos de la facultad, para ello se debe realizar entrevistas a todos los involucrados en los proyectos (Ver anexo # 6) con el objetivo de preparar un informe que contenga el perfil de madurez de cada proyecto. Se recomienda que estas entrevistas deban ser dirigidas a personas que tengan experiencias en CMMi y de ser posible en TMMi.

Toda la información obtenida se presenta al área de gestión de proyectos.

El informe debe contener la siguiente información:

Polo: \_\_\_\_\_

## Capítulo 2: Solución Propuesta

---

Proyecto: \_\_\_\_\_

Áreas de proceso	Puntuación	Conclusión
Estrategia de Pruebas		
Plan de pruebas		
Seguimiento		
Diseño y ejecución de los Casos de Prueba		
Ambiente de prueba		

**Tabla 1 Resultados de las entrevistas.**

\*La puntuación debe ser un número del 0 al 10

\*Las conclusiones deben de ser según la siguiente escala:

-Crítica (0-2), Deficiente (3-4), Regular (5-7), Adecuada (7-8) y Bueno (9-10).

- Definir los objetivos del proyecto de mejora los mismos deben de estar alineados con los objetivos que se traza la facultad. Para poder tener un mayor apoyo y compromiso de todos los involucrados en el proyecto.

### **Fase de Planificación.**

En esta fase se define como va a estar estructurado el proyecto de mejora en términos de estructura y los recursos que van a estar destinados al mismo.

- Se debe recibir toda la atención y los recursos necesarios para que el proyecto de mejora marque la diferencia.
- Mantener una adecuada comunicación para ello es muy importante brindar una información clara y efectiva del proyecto de mejora. Para que las personas vean que es muy importante el cambio y necesario para mejorar la calidad del proceso de pruebas. Para ello es conveniente enviar periódicamente correos informativos o boletines sobre el avance del proyecto de mejora

y los cambios en el mismo. Una buena práctica es la participación de estas personas en la revisión de los entregables del proyecto.

- Apoyarse en consultores externos. Los consultores externos a través de sus conocimientos y experiencia pueden hacer una valiosa contribución al proyecto de mejora especialmente si son asignados a tiempo completos. Los mismos permiten estimular las mejoras de proceso en términos de esfuerzo y progreso.
- Crear un compromiso con los interesados.
- Asegurarse de los procesos que dependen las pruebas. En este modelo no es solo importante observar la madurez del proceso de prueba, sino también los procesos de los que dependen las pruebas. Es necesario asegurarnos que la gestión de proyectos, gestión de la configuración, gestión de requisitos y gestión de riesgos tienen la suficiente atención en la facultad. La falta de una adecuada gestión de proyectos hará que el proyecto de mejora de pruebas sea un fracaso.
- Asegurarnos no iniciar un proyecto de mejora del área de pruebas si la facultad no está lista para soportarlo.
- Alinearse con otras iniciativas de mejora. Nuestra facultad puede estar implantando la mejora de procesos de software que propone CMMI por lo que hay que tener en cuenta que un proyecto productivo solo puede manejar una cierta cantidad de cambios en un determinado momento. Para evitar duplicar actividades que se ejecutan en un proceso de mejora, se recomienda tener un mapa de actividades exclusivo para la mejora de prueba. Esta cartografía debe ser responsabilidad de los jefes de proyecto.

### **Fase de Implementación.**

Durante esta fase, todos los cambios planeados deben ser realizados, especialmente aquellos que se obtuvieron de las recomendaciones rápidas de mejora.

- Aplicar Métricas. Con la utilización de métricas se mide el progreso y la calidad de las pruebas así como los establecimientos de indicadores de rendimiento para el proceso de pruebas. Las métricas son muy importante para controlar la calidad y para mostrar los beneficios del proceso de mejora.
- Detallar la documentación de forma clara y concisa. La misma debe definirse al inicio de la etapa de implementación.

- Definir la Política de Prueba.
- Definir una estrategia de prueba o plan de pruebas. Esta estrategia proporciona una visión de cómo los productos son probados a lo largo de su ciclo de vida. Deben evitarse probar los productos que se escapen del plan de pruebas, ya que al no tener una visión clara de lo que se hace, se corre el riesgo de perder el control del proyecto.

Todo plan de pruebas debería considerar aspectos como:

- Cuál es el objetivo de las pruebas.
- Cuál es el alcance de las pruebas.
- Que tipos de pruebas se realizaran.
  - ❖ Funcionales.
  - ❖ Stress.
  - ❖ Automatizadas.
  - ❖ De Regresión.
- Cuantos ciclos o barridas de prueba se estima realizar.
- Casos de prueba que se utilizaran.
- Que recursos realizarán las pruebas.
- Tiempo.

### **Fase de despliegue.**

- Asegurar la Accesibilidad de los entregables. Asegúrese de que todos los documentos, tanto los procedimientos y las plantillas, sean de fácil acceso para todos los probadores y otras partes interesadas.
- Dar a conocer los cambios y los resultados de la mejora del proceso de prueba.

### **2.6 Propuesta de capacitación del personal.**

La aplicación de la estrategia propuesta debe ser desarrollada por personas capacitadas en el modelo TMMI o que tengan conocimientos en CMMi debido a que la mayoría de los estudiantes de la facultad y los líderes de proyecto no están capacitados, es necesario ofrecer servicios de preparación en

diversos temas mediante un curso optativo para desarrollar y ampliar las capacidades de conocer el modelo, su estructura y aplicación.

### 2.5.1 Curso Optativo Introducción al TMMi.

Datos Generales	
Disciplina	Ingeniería y Gestión de Software.
Curso Optativo	Introducción al TMMi.
Perfil	Calidad
Año	2do Año, 3er Año, 4to Año y 5to Año

Tabla 2 Datos Generales del curso de capacitación.

### Objetivos Generales

#### Objetivos Instructivos:

- Conocer los conceptos, terminología y la estructura interna de TMMI.
- Describir los componentes de los modelos TMMI y sus relaciones.
- Discutir las áreas de procesos del TMMI.
- Entender el uso de TMMI en la operación y soporte del proceso de prueba dentro desarrollo de software.
- Localizar información relevante en el modelo.
- Comprender la aplicación del modelo TMMI en organizaciones y la influencia del mismo en la mejora del proceso de prueba.
- Interpretar adecuadamente el TMMI, en particular para aquellos que están involucrados en la mejora del proceso de pruebas.

#### Objetivos Educativos:

- Capacitar al personal sobre el Modelo integrado de Madurez de Prueba.
- Mejorar la estrategia para aplicar el framework TMMi en los proyectos de la facultad.

## Descripción de los Temas

### Tema 1: Modelo TMMI: Conceptos Básicos

- Antecedentes.
- Introducción al Modelo TMMI.
- Ámbito de Aplicación
- Estructura del Modelo.

### Tema 2: Modelo TMMI: Niveles de Madurez.

- Niveles de Madurez.

### Tema 3: Componentes del Modelo TMMI

- Áreas de procesos
- Metas específicas y genéricas
- Prácticas específicas y genéricas
- Subprácticas
- Productos típicos de trabajo
- Ejemplos

### Tema 4: Modelo TMMI. Nivel 2 de TMMI Gestionado.

- Características del nivel 2
- Áreas de proceso involucradas.
- Metas y Prácticas (específicas y genéricas) para alcanzar el nivel 2.

### Tema 5: Análisis de la estrategia para aplicar el modelo a los proyectos de la facultad.

- Objetivo de la estrategia.

## Capítulo 2: Solución Propuesta

---

- Descripción general de la estrategia.
- Propuestas de Mejora para la estrategia.

### Sistema de conocimientos:

Este curso se enfoca en el modelo TMMI y su representación, cubriendo los diferentes niveles de madurez y las áreas de procesos que lo componen, proporcionando una visión integral del modelo TMMi.

### Sistema de Evaluación de la asignatura

La evaluación durante el curso se realizará en diferentes escalas temporales:

- Evaluación periódica: El profesor es el encargado de realizar evaluaciones con el objetivo de ir probando que nivel de conocimiento están alcanzando los estudiantes, las mismas pueden ser a partir de una evaluación escrita individual sobre el tema tratado en la clase anterior.
- Trabajo final: Se orientará un trabajo investigativo que deben defender al final del curso. Dicho trabajo profundizará el estudio del las prácticas del Modelo TMMI y se entregará un informe escrito.

### Bibliografía

- “El Proceso Unificado de Desarrollo de Software”. Iver Jacobson, Grady Booch y James Rumbaugh. Addison-Wesley. 2000.
- Pressman, R. (1998). "Ingeniería de software. Un enfoque práctico."
- Erik van Veenendaal (2008). "Test Maturity Model Integration".
- Ing. Alexander Oré (2009). "TMMi cuando CMMi no es suficiente para la etapa de Pruebas". Disponible en: <http://www.calidadyssoftware.com>
- Sitio Oficial del TMMi Disponible en: <http://www.tmmifoundation.org>
- Tesis “Propuesta de adopción del framework TMMi como modelo de referencia para el proceso de pruebas en los proyectos productivos de la facultad 9” .Hector M Blanco Meriño (2009).



Elaborado	
Autor	Hector Manuel Blanco Meriño
Cargo	Estudiante del Proyecto de Calidad
Fecha	17 de mayo del 2009

Tabla 3 Datos del autor del curso de Capacitación.

### 2.7 Conclusiones Parciales.

En este Capítulo se hizo un estudio detallado del framework Modelo integrado de Madurez de Prueba. Mediante estudios realizados y síntesis de la información se conoce quien desarrolla el framework, se conoce sus antecedentes e historia. Su estructura que es similar a la que propone CMMI donde se hace una descripción detallada de cada uno de sus componentes para un mayor entendimiento. Se elabora una estrategia para aplicar el framework TMMi a los proyectos de la facultad donde se desglosa por pasos como aplicar un proyecto de mejora. Se propone un curso de capacitación ya que uno de los aspectos principales para aplicar el proyecto de mejora es que el personal esté capacitado en TMMi.

## Capítulo 3

### Capítulo 3: Evaluación de la propuesta.

#### 3.1 Introducción

En la investigación, hasta el momento, se realizó un estudio de las características del Modelo Integrado de Madurez de Prueba (TMMi, sus siglas en inglés) y sus niveles de madurez, se analizó su estructura y la importancia de aplicar este modelo. Se realizó una estrategia de cómo aplicar este modelo a los proyectos de la Facultad 9 y luego se hizo un estudio de cómo lograr satisfacer un área de proceso del nivel 2 de TMMi mediante las propuestas de metas específicas y prácticas específicas. Se hizo una propuesta de capacitación del personal con el objetivo de preparar a las personas para que tengan mayor conocimiento del tema tratado. En el presente capítulo para la validación y aceptación de todo lo planteado en el Capítulo 2, se tomó como herramienta el uso del Criterio de un Panel de Expertos y el empleo de técnicas propuestas en el Método de la Preferencia, y se realiza una descripción de cómo fue ejecutado el método y los resultados que fueron obtenidos.

#### 3.2 Método de Evaluación de Expertos.

Los métodos de expertos utilizan como fuente de información un grupo de personas a las que se supone un conocimiento elevado de la materia que se va a tratar. El principal objetivo de utilizar los métodos de expertos es buscar experiencia y criterios de un grupo de personas consideradas como expertas. Un experto según Durand(**DURAND, 1971**) es un individuo en sí, grupo de personas u organizaciones capaces de ofrecer valoraciones conclusivas de un problema en cuestión y hacer recomendaciones respecto a sus momentos fundamentales con un máximo de competencia. El lema de la evaluación de expertos es: "DOS CABEZAS PIENSAN MAS QUE UNA" (**ZATSIORSKI, 1989**).

Existen desde el punto de vista cualimétrico (estado de opinión) varios métodos de evaluación de expertos, tales como: Método de la comparación por pares, Método Delphi, Método de la preferencia.

El Método de la Preferencia es el más empleado, por su exactitud, objetividad y rapidez. Permite superar las limitaciones, relacionadas con la complejidad de su aplicación y del procesamiento de los datos y alcanzar una imagen integral y más amplia de la posible evolución del resultado científico

## Capítulo 3: Validación de la Solución Propuesta

---

sometido a valoración, reflejando las valoraciones individuales de los expertos, las cuales podrán estar fundamentadas, tanto en un análisis estrictamente lógico como en su experiencia intuitiva, y a la vez facilita el correspondiente análisis estadístico(FERNÁNDEZ, 2006).

Entre las etapas principales de realización de la evaluación de una investigación a través del método de evaluación de expertos se encuentran las siguientes:

### 3.2.1 Elaboración del objetivo.

Formulación del objetivo de la evaluación por los expertos: Valorar la propuesta de adopción del framework TMMi como modelo de referencia para el proceso de pruebas en los proyectos productivos de la facultad 9 y la efectividad que se espera alcanzar con su aplicación.

### 3.2.2 Selección de los expertos.

Se seleccionaron 7 expertos, tomando como criterio de selección la efectividad de la actividad profesional que realizan, la experiencia que poseen, por su reconocido prestigio profesional, los años vinculados a la UCI y su elevado conocimiento en ingeniería y gestión de software (Ver anexo # 4).

Elegir los expertos atendiendo a las características mencionadas propicia obtener resultados con calidad, junto a otras cualidades propias de éstos como pueden ser: la seriedad, la honestidad, la sinceridad, la responsabilidad y otras en este sentido, que hacen que las opiniones brindadas sean confiables y válidas para el objetivo propuesto.

### 3.2.3 Elaboración del cuestionario.

En la confección de la encuesta se tuvieron en cuenta los aspectos principales que debe tener la propuesta para su aplicación efectiva en el proceso productivo de la facultad 9. El cuestionario está compuesto por 4 preguntas con un enfoque investigativo, brindando la posibilidad de que cada experto dé su opinión de manera anónima y a su vez abierta, proporcionando riqueza en sus respuestas (Ver anexo# 5).

### 3.2.4 Elección de la metodología.

La evaluación de los expertos se desarrolló a través del Método de la Preferencia ya que es el método más empleado, por su exactitud, objetividad y rapidez. Permite superar las limitaciones, relacionadas con la complejidad de su aplicación y del procesamiento de los datos y alcanzar una imagen integral y más amplia de la posible evolución del resultado científico sometido a valoración, reflejando las

## Capítulo 3: Validación de la Solución Propuesta

---

valoraciones individuales de los expertos, las cuales podrán estar fundamentadas, tanto en un análisis estrictamente lógico como en su experiencia intuitiva, y a la vez facilita el correspondiente análisis estadístico.

### 3.2.5 Ejecución de la metodología.

La ejecución de la metodología se inicia con la entrega a cada experto de la encuesta con los aspectos a evaluar, donde los mismos deben expresar sus ideas y criterios sobre la solución propuesta, las mejoras, deficiencias que presenta la solución y los que representara su aplicación en los proyectos de la facultad 9. La misma es mandada por correo a cada experto.

### 3.2.6 Procesamiento de la información.

El 100% de los expertos considera que solución propuesta está a la altura de las necesidades y posibilidades de aplicación para la mejora del proceso de pruebas en los proyectos productivos de la facultad 9.

El experto 2 plantea que sería de gran significación implantar TMMi al proceso de pruebas de la facultad 9, ya que el modelo es algo novedoso y que va encaminado a la mejora de procesos de prueba basados en los niveles de madurez, el mismo mejora grandemente los procesos de pruebas provocando un gran impacto en la calidad del producto final.

Los expertos 1 y 6 plantean que el modelo propuesto es de suma importancia ya que nuestra facultad al igual que la universidad se encuentra enfrascada en la mejora de sus procesos con la utilización del CMMI pero el CMMi deja algunas brechas en las áreas de las pruebas las cuales la solución planteada le da solución.

El 100% de los expertos plantean que la solución propuesta podrá erradicar los problemas que existen en el proceso de pruebas.

Los expertos 4,2 y 7 plantean que si se logra mejorar el proceso de pruebas al producto que realice la facultad en sus proyectos productivos tendrá una menor probabilidad de tener problemas de aceptación del cliente y ya que los errores detectados serán en etapas tempranas y su solución será menos costosa y por tanto la efectividad en el trabajo sería mejor.

Los expertos 3 y 5 plantean que si se logra establecer bien la política de prueba y se confecciona bien una buena estrategia de pruebas esto podrá guiar bien el proceso el proceso de mejora y así se

## Capítulo 3: Validación de la Solución Propuesta

ganara en tiempo y se aumentarían la calidad del producto final ya que se podrán descubrir errores en etapas iniciales del proyecto.

El 100 % de los expertos consideran que la adopción de framework TMMI como modelo de referencia para la mejora del proceso de pruebas sea suficientemente factible a las necesidades de los proyectos productivos de la facultad ya que el mismo proporciona un enfoque más amplio y estructurado al proceso de prueba.

La última pregunta de la encuesta se basa en valorar la propuesta según ciertos criterios establecidos (criterios de mérito científico, implantación, generalización e impacto) dándole valores en una escala del 1 al 5, donde 5 es la máxima evaluación. Los criterios establecidos son:

1. Satisfacción a las necesidades de los proyectos productivos.
2. Calidad de la investigación.
3. Novedad científica.
4. Aporte científico.
5. Facilidad de comprensión.
6. Adaptabilidad a diferentes entornos de producción de software.
7. Repercusión en los proyectos productivos.
8. Contribución al proceso de desarrollo de software.

Los resultados emitidos por los expertos según los criterios que se plantean, se muestran en la siguiente tabla:

Expertos	Criterios	1	2	3	4	5	6	7	8
	1		4	5	5	4	5	5	4
2		5	4	4	3	4	5	5	4
3		3	4	3	3	4	3	5	4
4		4	5	4	4	4	3	3	4

## Capítulo 3: Validación de la Solución Propuesta

5	4	5	4	4	5	5	4	5
6	4	4	4	4	4	4	4	4
7	4	5	5	5	5	3	5	4

Tabla 4 Resultados de los expertos según los criterios establecidos en la pregunta 5.

A partir de los resultados que muestra la Tabla 5, se crea la tabla de los rangos de puntajes ligados:

Expertos	Criterios	1	2	3	4	5	6	7	8
	1		2.5	6.5	6.5	2.5	6.5	6.5	2.5
2		7	3.5	3.5	1	3.5	7	7	3.5
3		2.5	6	2.5	2.5	6	2.5	8	6
4		5	8	5	5	5	1.5	1.5	5
5		2.5	6.5	2.5	2.5	6.5	6.5	2.5	6.5
6		4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5
7		2.5	6	6	6	6	1	6	2.5
Sj		26.5	41.0	30.5	24.0	38.0	36.0	32.0	30.5

Tabla 5 Rangos Porcentuales según los criterios de la pregunta 5

### Cálculo de la media de los rangos.

Con los resultados obtenidos en la tabla 5 de rangos porcentuales sumo todos los  $R_j$  y esa suma me da el valor de  $S_j$  y este lo divido entre la cantidad de criterios tratados  $n$ .

$$S^- = \frac{\sum_{j=1}^n S_j}{n} = \frac{258.5}{8} = 32.3125$$

### Cálculo de Suma de cuadrados de las desviaciones de sumas

## Capítulo 3: Validación de la Solución Propuesta

$$S = \sum_{j=1}^n (S^- - S_j)^2$$
$$= (32.31 - 26.5)^2 + (32.31 - 41.0)^2 + (32.31 - 30.5)^2 + (32.31 - 24.0)^2$$
$$+ (32.31 - 38.0)^2 + (32.31 - 36.0)^2 + (32.31 - 32.0)^2 + (32.31 - 30.5)^2 = 230.996$$

### Cálculo de Factor de Corrección para cada experto.

Donde  $t$  son las veces que se repiten los números en la tabla de rangos de derecha a izquierda. Los valores que no se repiten no se tienen en cuenta.

1.  $T_i = \frac{\sum_{j=1}^n (t^3 - t)}{12} = \frac{((4^3 - 4) + (4^3 - 4))}{12} = \frac{60 + 60}{12} = 10$
2.  $T_i = \frac{\sum_{j=1}^n (t^3 - t)}{12} = \frac{((4^3 - 4) + (3^3 - 3))}{12} = \frac{60 + 24}{12} = 7$
3.  $T_i = \frac{\sum_{j=1}^n (t^3 - t)}{12} = \frac{((3^3 - 3) + (4^3 - 4))}{12} = \frac{24 + 60}{12} = 7$
4.  $T_i = \frac{\sum_{j=1}^n (t^3 - t)}{12} = \frac{((5^3 - 5) + (2^3 - 2))}{12} = \frac{120 + 6}{12} = 10.5$
5.  $T_i = \frac{\sum_{j=1}^n (t^3 - t)}{12} = \frac{((4^3 - 4) + (4^3 - 4))}{12} = \frac{60 + 60}{12} = 10$
6.  $T_i = \frac{\sum_{j=1}^n (t^3 - t)}{12} = \frac{((8^3 - 8))}{12} = \frac{504}{12} = 42$
7.  $T_i = \frac{\sum_{j=1}^n (t^3 - t)}{12} = \frac{((2^3 - 2) + (5^3 - 5))}{12} = \frac{6 + 120}{12} = 10.5$

### Cálculo de la suma de todos los Factores de Correlación.

Donde  $m$  es el número de expertos.

$$\sum_{i=1}^m T_i = 97$$

### Cálculo Coeficiente de Concordancia de KENDALL.

Donde  $m$  es el número de expertos,  $n$  el número de criterios tratados.

$$W = \frac{12 S}{m^2(n^3 - n) - m \sum_{i=1}^m T_i} = \frac{2771.952}{24017} = 0.1154$$

# Capítulo 3: Validación de la Solución Propuesta

---

## **Prueba de Significación para el W.**

$$X^2 = m(n - 1)W = 5.6546 \text{ (Real)}$$

Se calcula la diferencia y se busca en la tabla de probabilidad  $X^2$  (Ver Anexo # 7), obteniéndose:

$$df = n - 1 = 7; \quad X^2_{(0.001,7)} = 24.32$$

## **Error de la Evaluación.**

Como se seleccionaron 7 expertos, el error de la evaluación es del 10%, es decir, la confiabilidad del criterio emitido por los expertos será del 90.0%.

## **Conclusión**

Como  $X^2_{real} < X^2_{(0.001,7)}$  entonces, los resultados de la evaluación de la propuesta de adopción del framework TMMi como modelo de referencia para el proceso de pruebas en los proyectos productivos de la facultad 9, realizada por los expertos, son de significación estadística, es decir, hay evidencias suficientes para plantear, a un 99.9% de confiabilidad, que los 7 expertos concuerdan en la efectividad de la propuesta de adopción del framework TMMi como modelo de referencia para el proceso de pruebas en los proyectos de la facultad 9, tanto en su concepción teórica como en los resultados que se obtendrán con su aplicación en los proyectos de la facultad 9.

## **3.3 Análisis de beneficios y costo.**

El modelo TMMi es un modelo de referencia para la mejora del proceso de prueba que orienta a todos los desarrolladores de software como hacer más efectivo el proceso de pruebas y obteniéndose grandes resultados en la calidad del producto final. La meta principal es lograr una mayor calidad en el desarrollo de soluciones informáticas. La calidad de un producto es determinada en gran medida por la calidad del proceso de prueba utilizado.

La aplicación de un modelo de estas características es un proceso largo y costoso que puede tardar hasta 2 años para pasar de un nivel a otro, por lo que implica un mayor esfuerzo. Los beneficios que va a obtener la facultad con la utilización de este modelo de referencia para la mejora del proceso de prueba serán mucho entre los que se pueden señalar:

Una reducción de costos:



## Capítulo 3: Validación de la Solución Propuesta

---

- Una mayor fiabilidad de las planificaciones de las pruebas y su desarrollo.
- Reducción de defectos.
- Mayor aceptación del producto por parte del cliente ya que el producto cumplirá las exigencias puestas por el cliente.

Aumento en la confiabilidad por:

- Reducción consistente de errores (reduciendo el número de defectos y detección en las fases más tempranas del ciclo de vida).
- Cumplimiento de fechas ya que el producto desarrollado cumplirá las exigencias puestas por el cliente.

Mayor efectividad por:

- Visibilidad sobre el proceso y sobre el producto.
- Operar con estándares documentados.
- Personal formado.

Mejorará en la imagen del producto desarrollado ya que se entregará un producto de mayor calidad.

El empleo de Modelo Integrado de Madurez de Prueba le permitirá a la facultad obtener los siguientes beneficios:

- Dar soporte necesario a las actividades de pruebas que realiza la facultad en los proyectos potenciando una mejora del proceso de pruebas. Contribuyendo a desarrollar un producto con mayor calidad.
- Mejorar el proceso de prueba con la utilización del framework TMMi.
- Ampliar el alcance y la visibilidad dentro del ciclo de vida del producto para permitir que el producto responda a las expectativas del cliente.
- Resaltar el desarrollo de la mejora del proceso de prueba en la facultad que permitan mejorar el desarrollo de los productos y los servicios ofertados a los clientes.

## *Capítulo 3: Validación de la Solución Propuesta*

---

### **3.4 Conclusiones Parciales.**

La evaluación de la Propuesta de adopción del framework TMMi como modelo de referencia para el proceso de pruebas en los proyectos productivos de la facultad 9, se realizó por 7 expertos seleccionados según la efectividad de la actividad profesional que realizan y mediante el empleo de técnicas del Método de la Preferencia. Esta evaluación permite concluir que con la adopción del framework se mejorará el proceso de prueba en la facultad lográndose obtener en la misma un producto con mayor calidad. Se realizó una propuesta de metas y prácticas específicas de para alcanzar el nivel 2 de TMMi en el área de proceso Política de Prueba y Estrategia, así como una caracterización de las demás áreas. Por último se realizó un análisis del costo-beneficio que resultaría al aplicar el Modelo TMMI.

## Conclusiones Generales

Con la realización de este trabajo se logró hacer un estudio del Modelo Integrado de Madurez de Prueba (TMMi, sus sigla en inglés) llegándose a la conclusión que el mismo debe ser adoptado por la facultad 9 para la mejora de su proceso de pruebas ya que este modelo proporciona un enfoque más amplio y estructurado al proceso de prueba. El mismo define claramente 16 áreas de procesos, que a través de metas y prácticas (específicas y genéricas) van definen el nivel de madurez de una institución.

Con la realización de este trabajo se puedo llegar a la siguiente conclusión:

- La calidad de los productos desarrollados en la facultad dependerá mucho de las pruebas de software realizadas y a su vez de la mejora que se le realice a este proceso de pruebas. Ya que los productos desarrollados cumplirán con las expectativas de los clientes, se incurrirán en menos defectos y serán más predecibles.
- Si se mejora el proceso de pruebas en la facultad los productos desarrollados tendrán una mayor aceptación por parte de los clientes.
- El proceso de pruebas de la facultad será planificado y ejecutado según la política de prueba establecida por la facultad, involucrará a los principales interesados en la mejora del proceso de pruebas, será monitoreado, controlado y revisado continuamente.

## Recomendaciones

Después de analizar los beneficios del uso que traerá el modelo TMMi para la facultad y su contribución a la mejora del proceso de pruebas se recomienda:

- Continuar el estudio de Modelo Integrado de Madurez de Prueba ya que es un modelo novedoso y está desarrollándose todavía.
- Continuar estudiando las áreas de proceso del Nivel 2 ya que es el nivel más próximo a alcanzar en nuestra Facultad.
- Utilizar la presente propuesta como base inicial para lograr una motivación en las personas a que usen el modelo para mejorar el proceso de pruebas.

## Referencias Bibliográficas.

- AGUILAR, J. M. *Otras 101 citas célebres del mundo de la informática* [Consultado el: abril de 2008]. Disponible en: [http://www.variablenotfond.com/2008\\_04\\_01\\_archive.html](http://www.variablenotfond.com/2008_04_01_archive.html).
- BEIZER, B. *Software Testing Techniques*. Editado por: Reinhold, V. N. 1990.
- CALDAS, U. D. F. J. *VISIÓN GENERAL DE LAS PRUEBAS DE SOFTWARE*  
Bogota Disponible en: <http://www.udistrital.edu.co/comunidad/grupos/arquisoft/index.php?id=40&type=1>.
- CÁRDENAS, M. C. C. *Pruebas de software ¿la salvación, un proceso sin utilidad, trivial, simplemente una moda, o...?* publicado el: febrero 2009 de 2007, última actualización: febrero 2009. Disponible en: [http://www.acis.org.co/fileadmin/Base\\_de\\_Conocimiento/XXVII\\_Salon\\_Informatica/MariaClaraChoucair-PruebasDeSoftware.pdf](http://www.acis.org.co/fileadmin/Base_de_Conocimiento/XXVII_Salon_Informatica/MariaClaraChoucair-PruebasDeSoftware.pdf).
- CARRASCO, O. F.; LEÓN, D. G., *et al. Un enfoque actual sobre la calidad del software* La Habana ACIMED, [Consultado el: Febrero de 2009]. Disponible en: [http://www.bvs.sld.cu/revistas/aci/vol3\\_3\\_95/aci05395.htm](http://www.bvs.sld.cu/revistas/aci/vol3_3_95/aci05395.htm).
- DURAND, R. *El método Delphy y la perspectiva del Hidrogeno* Madrid, 1971, vol. 11, nº
- ESPINOZA, D. R. V. y ESPINOZA, E. G. V. *Estándar de calidad para pruebas de software* Investigativa, Nacional Mayor de San Marcos 2005.
- FERNÁNDEZ, R. *Los métodos de evaluación de expertos para valorar resultados de las investigaciones* 2006.
- GELPERIN, D. y HETZEL, B. *The Growth of Software Testing*  
CACM, 1998 vol. 31, nº 6, Disponible en: <http://portal.acm.org/citation.cfm?id=62959.62965>.
- GUERRA, R. T. y OCA, J. R. *Calidad en el desarrollo de software* publicado el: febrero de 2005, última actualización: febrero Disponible en: [http://fenix.utim.edu.mx/man/general/biblioteca/info2004/Calidad\\_Desarrollo\\_Software.pdf](http://fenix.utim.edu.mx/man/general/biblioteca/info2004/Calidad_Desarrollo_Software.pdf).
- HUACOTO, N. *Propuesta para implantar CMMI en una empresa con múltiples unidades desarrolladoras de software*. Universidad Nacional Mayor de San Marcos, 2005.
- IEEE\_829. *Standard for Software Test Documentation*. IEEE Standards Board, 1998,
- ISO8402. *Gestión de la Calidad y Aseguramiento de la Calidad*. 1994,
- ISO9000. *Estándar de Gestión de Calidad*. 2000, nº
- JACOBSON, I.; BOOCH, G., *et al. El proceso Unificado de desarrollo de Software* Madrid: Pearson Educación 2000.

- JSI. Japanese Industrial Standard JIS z8110-1981. 1981, nº
- KOOMEN, T. y POL, M. *Test Process Improvement*. Addison Wesley, 1999
- LOVELLE, J. M. C. *Calidad del Software* En 21 Octubre 1999.1999.
- LLEBREZ, V. R. M. y CASTILLO, L. A. S. *Concepción de la Calidad en el pensamiento del Ché* Ciencias Sociales ed. La Habana: 2006.
- MÓNOCA, S. D. y VITALI, S. *El modelo IDEAL para implementar CMMI*. En *El modelo IDEAL para implementar CMMI*. 12 de diciembre 2005.
- MONTOYA, E. R. *Pruebas de Software UNAD Cead Tunja EvalSoftware* [Consultado el: enero de 2009 ].  
Disponible en: <http://rolandounadevalsoftware.blogspot.com/>.
- MUTAFELIJA, B. *Systematic Process Improvement using ISO 9001:2000 and CMMI*. Artech House Computing Library. 2003, nº
- ORÉ, I. A. *TMMI cuando el CMMI no es suficiente* Disponible en:  
[http://www.calidadyssoftware.com/testing/tmmi\\_nuevo\\_cmmi\\_enfocado\\_a\\_pruebas.php](http://www.calidadyssoftware.com/testing/tmmi_nuevo_cmmi_enfocado_a_pruebas.php).
- PRESSMAN, R. *Ingeniería de Software :Un Enfoque Práctico*. La Habana: Editorial Félix Varela, 2005.
- ROMERO, C. I. G. *El Modelo de Capacidad de Madurez y su Aplicación a empresas Mexicanas de Software* Guadalajara Disponible en: [http://mailweb.udlap.mx/~tesis/lis/garcia\\_r\\_ci/](http://mailweb.udlap.mx/~tesis/lis/garcia_r_ci/)
- RUBIO, G. B. *Calidad en Ingeniería de Software* Universidad de las Illes Balears, [Consultado el: Febrero de 2009].  
Disponible en: <http://dmi.uib.es/~bbuades/calidad/sld001.htm>.
- SÁNCHEZ, P. F. *Mejora de procesos para desarrollar Software Mejor* Colombia [Consultado el: febrero de 2009].  
Disponible en: <http://www.slideshare.net/pfsanchez/mejora-de-procesos-para-desarrollar-software-mejor>.
- SANZ, D. D. L. F. *TestPAI:Un área de proceso de pruebas integrada con CMMi*. diciembre 2008, vol. 4, nº 4,  
Disponible en: <http://www.ati.es>.
- SOFTWARE, C. D. *Fundamentos* Guadalajara [Consultado el: 21 febrero de 2009]. Disponible en: <http://www.e-quality.com.mx/definiciones.php>.
- SOMERVILLE, I. *Ingeniería de Software*. Addison Wesley 2002.
- VEENENDAAL, E. V. *Test Maturity Model Integration*. TMMi Foundation, 2008.
- VEENENDAAL, E. V. y TEUNISSEN, R. *Software Testing, A guide to the TMap Approach*. Addison Wesley, 2002.

VILLA, M. D. L.; RUIZ, I., *et al.* *Modelos de Evaluación y Mejora de Procesos: Análisis Comparativo*. publicado el: febrero 2009 de 2004, última actualización: febrero 2009. Disponible en: <http://www.sc.ehu.es/jiwdocoj/remis/docs/DelaVillaadis2004.doc>.

WESLEY, A. y SOMERVILLE, I. *Ingeniería de Software*. 2002.

ZAMURIANO, R. F. *Las Inspecciones de Software y las Listas de Comprobación*. 2000.

ZATSIORSKI, V. *Métodología Deportiva*. La Habana: Editorial Pueblo y Educación 1989.

## Bibliografía Consultada

- Cárdenas, Maria Clara Choucair. 2007.** Pruebas de software ¿la salvación, un proceso sin utilidad, trivial, simplemente una moda, o... ? [En línea] 2007. [Citado el: 3 de febrero de 2009.] [http://www.acis.org.co/fileadmin/Base\\_de\\_Conocimiento/XXVII\\_Salon\\_Informatica/MariaClaraChoucair-PruebasDeSoftware.pdf](http://www.acis.org.co/fileadmin/Base_de_Conocimiento/XXVII_Salon_Informatica/MariaClaraChoucair-PruebasDeSoftware.pdf).
- Castillo, Vicente R. Martínez LLebrez Luís A Sabadí. 2006.** *Concepción de la Calidad en el pensamiento del Che*. La Habana : Ciencias Sociales, 2006.
- Dunaway, Marilyn Bush Donna. 2005.** *CMMI Assessments .Motivating Positive Change*. United States : Pearson Education, 2005.
- Fernández, Raúl. 2006.** *Los métodos de evaluación de expertos para valorar resultados de las investigaciones* . 2006.
- Gallego, Juan Pablo Gómez. 2007.** Fundamentos de la Metodología RUP . [En línea] 2007. <http://www.scribd.com/doc/297224/RUP>.
- Kruchten, Philippe. 2003.** *The Rational Unified Process An Introduction*. s.l. : Addison Wesley, 2003.
- López, Carlos. 2001.** Gestión Polis. *Gestión Polis*. [En línea] 11 de 2001. [Citado el: 08 de 01 de 2009.] <http://www.gestiopolis.com/canales/gerencial/articulos/27/asesis.htm>.
- Oré, Ing .Alexander. 2009.** TMMI cuando el CMMI no es suficiente . [En línea] 2009. [Citado el: 20 de enero de 2009.] [http://www.calidadyssoftware.com/testing/tmmi\\_nuevo\\_cmml\\_enfocado\\_a\\_pruebas.php](http://www.calidadyssoftware.com/testing/tmmi_nuevo_cmml_enfocado_a_pruebas.php).
- Piedra, Nelson. 2007.** Mejora de Procesos: Clave de la competitividad en el mercado global de TI. [En línea] 2007. <http://nopiedra.wordpress.com/category/spi/>.
- Pressman, Roger. 2005.** *Ingeniería de Software :Un Enfoque Práctico*. La Habana : Editorial Félix Varela, 2005.
- Ramos, Manuel de la Villa Isabel Ruiz Isabel. 2004.** Modelos de Evaluación y Mejora de Procesos: Análisis Comparativo. [En línea] 2004. <http://www.sc.ehu.es/jiwdocoj/remis/docs/DelaVillaadis2004.doc..>
- Rumbaugh, Ivar Jacobson Grady Booch James. 2000.** *El proceso Unificado de desarrollo de Software* . Madrid : Pearson Educación , 2000.



**Sánchez, Pablo Fernando. 2007.** Mejora de procesos para desarrollar Software Mejor . [En línea] 2007.  
<http://www.slideshare.net/pfsanchez/mejora-de-procesos-para-desarrollar-software-mejor>.

**Sanz, Luís Fernández. 2008.** La importancia de la calidad del software. [En línea] 2008.  
<http://www.baquia.com/articulos/software/noticia/14128/la-importancia-de-la-calidad-del-software>.

**Teruel, Alejandro. 2001.** Las Pruebas de Verificación de Requerimientos. [En línea] 2001.  
<http://www ldc.usb.ve/~teruel/ci4713/clases2001/testReqs.html>.

*TestPAI:Un área de proceso de pruebas integrada con CMMi.* **Sanz, Dr Luís Fernández. 2008.** 4, Madrid : s.n., 2008, Vol. 4.

**Veenendaal, Erik van. 2008.** *Test Maturity Model Integration.* s.l. : TMMi Foundation, 2008.

## **Glosario de Términos.**

**CMMI:** Modelo de Capacidad de Madurez Integrado.

**TMMi:** Modelo Integrado de Madurez de Prueba.

**CMM:** Modelo de Madurez de las Capacidades.

**TMM:** Modelo de Madurez de Prueba.

**SPI:** Mejora del Proceso de Software.

**Institucionalización:** Lograr cambiar los hábitos y la rutina diaria del trabajo a partir de un cambio en la cultura organizacional. La institucionalización de las prácticas es la única medida real de progreso. Definición y capacitación son avances parciales.

**ISO:** Organización Internacional de Estándares.

**IEEE:** Instituto de Ingenieros Eléctricos y Electrónicos

**SEI:** Instituto de Ingeniería de Software (Software Engineering Institute) de la Universidad Carnegie Mellon (EEUU). Autor de los modelos CMMI y del método SCAMPI.

**Proceso:** conjunto de pasos a seguir.

**Componente:** Pequeña parte del software que puede ser probado aisladamente.

**Defecto:** Anomalía del sistema de software.

**Modelo:** Indicar que lo que ha sido creado es ejemplar o se considera que puede serlo.

**Referencia:** Es toda la información sobre un tema determinado.

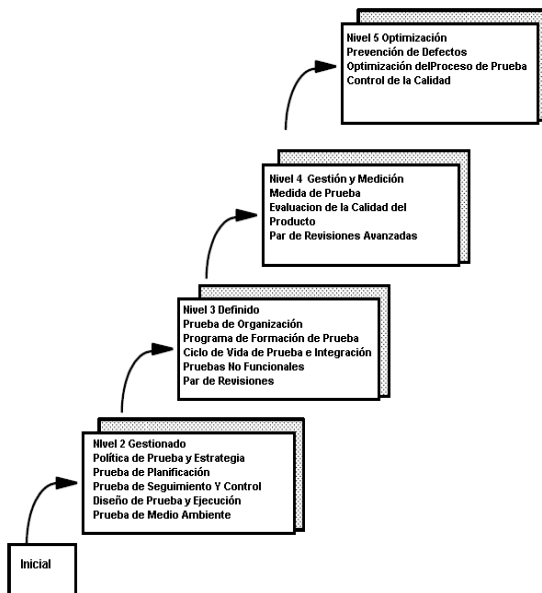
**Nivel:** Altura que algo alcanza, o a la que está colocado.

**Anexos.**

**Anexo # 1 Entrevista Realizada a los líderes de Proyecto.**

1. ¿Cómo realizan el proceso de desarrollo de software en su proyecto?
2. ¿Qué metodología es la que aplica su proyecto?
3. ¿Cómo está organizado el proyecto?
4. ¿Tienen el proyecto un grupo de aseguramiento calidad?
5. ¿Cómo se evalúa la calidad del producto?
6. ¿Cuáles son los principales problemas que se han presentado durante el desarrollo del producto?
7. ¿Realizan son las pruebas de software en su proyecto?
8. ¿Conoce los tipos de pruebas existentes?
9. ¿Cuáles se aplican en su proyecto?
10. ¿Cómo realizan las pruebas de software en su proyecto?
11. ¿Se documentan los defectos encontrados en esta fase de pruebas?
12. Realizan las pruebas durante todo el ciclo de desarrollo de software.

**Anexo # 2 Niveles de Madurez del TMMi con sus áreas de proceso.**



**Ilustración 1** Niveles de madures y áreas de proceso del TMMi tomado del libro Test Maturity Model Integration(TMMI) de Erik van Veenendaal (VEENENDAAL, ERIK VAN y TEUNISSEN, 2002).

Anexo # 3 Estructura del TMMI.

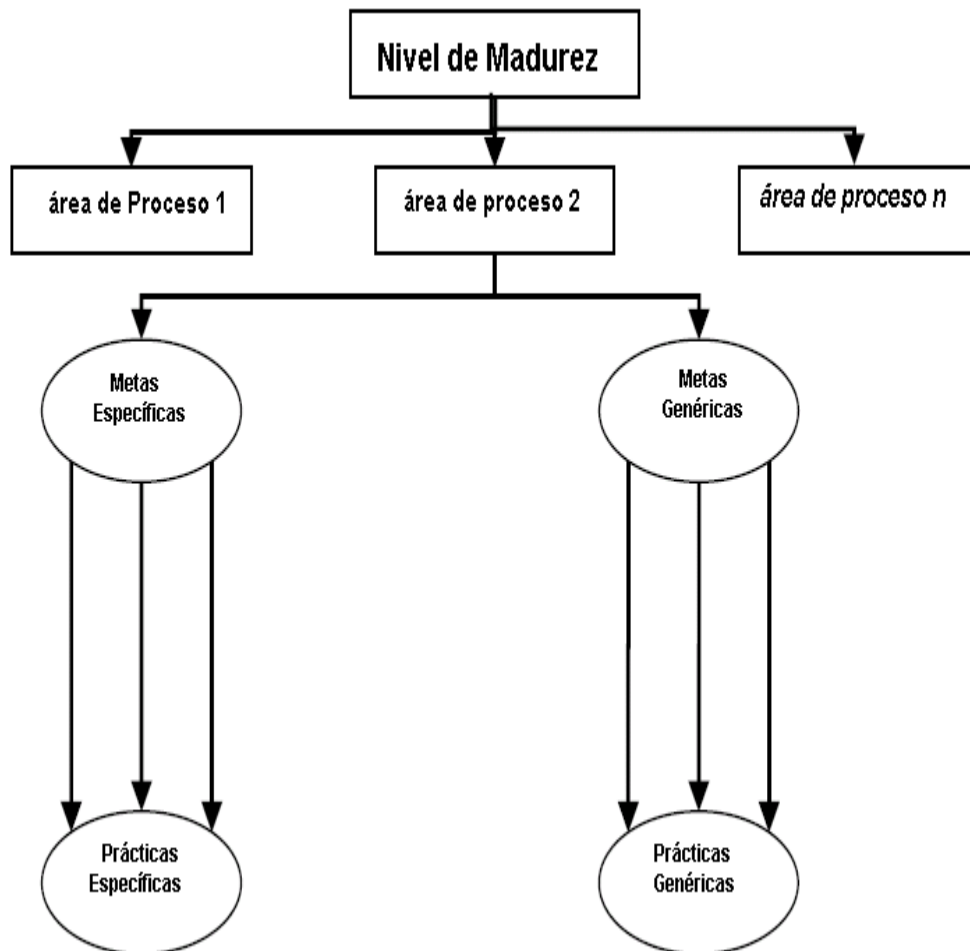


Ilustración 2 Estructura del TMMi tomada del libro TMMi Foundation de Erik van Veenendaal (VEENENDAAL, ERIK VAN y TEUNISSEN, 2002).

**Anexo # 4 Tabla que contiene los datos de los expertos.**

Vinculado a Proyecto Productivo. (rol)	Graduado De:	Años de graduado	Años vinculado a la UCI	Grado Científico	Eventos Científicos
Grupo de Auditorías y Revisiones (CALISOFT)	Ingeniero en Ciencias Informáticas	1	1	-	Fórum de ciencia y técnica. Jornada Científica Estudiantil
Analista Principal del Proyecto Manugas.	Ingeniero Informático	2	2	-	Fórum de ciencia y técnica. Jornada Científica Estudiantil
Jefa del equipo de calidad del polo de Geoinformática	Ing. Ciencias Informáticas	1	1	-	Fórum de Ciencia y Técnica
Líder de Proyecto MENPET y Primicia y Líder Polo Video y Sonido Digital	Ing. Ciencias Informáticas	2	2	-	UCIENCIA INFORMÁTICA 2007 y 2009
Grupo de Calidad del Polo Geoinformática.	Ing. Ciencias Informáticas	2	2	-	Jornada Científica Estudiantil

Jefe de Proyecto SIG-UCI	Ingeniero en Ciencias Informáticas	1	1		Fórum de Ciencia y Técnica  Jornada Científica Estudiantil
Proyecto Gestión de Costo .(DDC Ciencias empresariales)	Ingeniero en Ciencias Informáticas	1	1		Fórum de Ciencia y Técnica  Jornada Científica Estudiantil

### Anexo # 5 Encuesta realizada a los expertos.

#### Encuesta a los expertos

Usted ha sido seleccionado por su conocimiento en calidad y gestión de software, por sus años de experiencia en proyectos productivos y los resultados alcanzados en su labor profesional, como experto para evaluar los resultados teóricos de esta investigación.

1. ¿Considera usted que la propuesta está a la altura de las necesidades y posibilidades de aplicación de los proyectos productivos de la facultad 9?

Si\_\_\_ No\_\_\_ ¿Por qué?

2. ¿Con la propuesta establecida se podrá erradicar los problemas que existen en el proceso de pruebas que utilizan los proyectos, cree usted que se podrá aumentar la efectividad del trabajo?

Si\_\_\_ No\_\_\_ ¿Por qué?

3. ¿Considera usted que la adopción de framework TMMI como modelo de referencia para la mejora del proceso de pruebas sea suficientemente factible a las necesidades de los proyectos productivos?

Si\_\_\_ No\_\_\_ Si cree necesario que no va acorde con las necesidades, méncionelo y explique brevemente.

4. En una escala del 1 al 5 confiera una evaluación a la propuesta según los siguientes criterios:

\_\_\_ Satisfacción a las necesidades de los proyectos productivos.

\_\_\_ Calidad de la investigación.

\_\_\_ Novedad científica.

\_\_\_ Aporte científico.

\_\_\_ Facilidades de comprensión.

\_\_\_ Adaptabilidad a diferentes entornos de producción de software.

\_\_\_ Repercusión en los proyectos productivos.

\_\_\_ Contribución al proceso de desarrollo de software.

## **Anexo # 6 Cuestionario de preguntas para conocer el nivel en que se encuentran los proyectos.**

### **Administración de requerimientos:**

¿Cuál es el tratamiento que se aplica para gestionar los cambios?

¿Cómo se logra el acuerdo con el sector usuario una vez que se ha relevado y documentado el requisito?

¿Qué herramientas para gestionar el requisito se utiliza?

¿Cómo se desarrolla y mantiene la vinculación entre: requerimientos, casos de uso, configuración y diseño?

### **Administración de proyecto:**

¿Qué tipo de técnicas se aplican para estimar las tareas?

¿Cómo se planifica la estrategia de abordaje?

¿Cómo se definen los ciclos?

¿Cómo se reconocen los riesgos a gestionar?

¿Cómo se acuerda la planificación entre las partes involucradas (técnicas y usuarias)?

## **Administración de configuraciones:**

¿Cómo se administra la configuración?

¿Qué herramientas se utilizan?

¿Cómo y cuándo se generan las líneas bases?

¿Cómo y cuándo se generan los diferentes versionados?

## **Aseguramiento de la calidad:**

¿Cuál es el concepto que se maneja internamente? ¿Cómo lo aplican?

¿Cómo se asegura la calidad de la forma de trabajo?

¿Cómo se asegura la calidad del producto?

¿Cuál es la relación entre calidad y las pruebas de software?

## **Monitoreo y control de proyectos:**

¿Qué y Cuándo se monitorea?

¿Cómo se detectan, identifican y clasifican los defectos y errores?

¿Cuál es el seguimiento que se aplica hasta el cierre de los mismos?

## **Medida y Análisis:**

¿Qué se está midiendo y con qué objeto?

¿Cuál es la vinculación con las reglas de negocio?

¿Cómo se planifica la medición?

¿Cómo se toman los datos e información que servirán para medir?

¿Cómo se analizan los resultados de la medición?

¿Cuál es el tratamiento que se sigue de los resultados obtenidos?

## **Administración de proveedores:**

¿Cuál es tratamiento que se tiene con los proveedores?

¿Cuál es el criterio de selección?



¿Cuál es la política y procedimientos para alcanzar los acuerdos con ellos?

¿Cómo se administra la relación con ellos?

**Anexo # 7 Valores críticos de  $X^2$ .**

<b>Probabilidades</b>				
<b>Df</b>	<b>0.10</b>	<b>0.05</b>	<b>0.01</b>	<b>0.001</b>
<b>4</b>	7.78	9,49	13,28	18,46
<b>5</b>	9.24	11,07	15,09	20,52
<b>6</b>	10.64	12,59	16,81	22,46
<b>7</b>	12.02	14,07	18,48	24,32
<b>8</b>	13.36	15,51	20,09	26,12
<b>9</b>	14.68	16,92	21,67	27,88
<b>10</b>	15.99	18,31	23,21	29,59
<b>11</b>	17.28	19,68	24,72	31,36
<b>12</b>	18.55	21,03	26,22	32,91
<b>13</b>	19.81	22,36	27,69	34,53
<b>14</b>	21.06	23,68	29,14	36,12
<b>15</b>	22.31	25,00	30,58	37,70
<b>16</b>	23.54	26,30	32,00	39,29
<b>17</b>	24.77	27,59	33,41	40,75

Tabla 6 Valores críticos de  $X^2$ .

**Anexo # 8 Planilla que propone RUP para realizar una estrategia de prueba.**

**Estrategia de prueba**

**Versión <1.0>**

**Revisión Histórica**

<b>Fecha</b>	<b>Versión</b>	<b>Descripción</b>	<b>Autor</b>
<día/mes/año>	<#>	<detalles>	<nombre>

## Estrategia de Prueba

### 1. Introducción

#### 1.1 Propósito

*El propósito de la Estrategia de la Prueba para el <el ciclo de vida completo, una fase específica> del <el Nombre del Proyecto> es:*

- Proporcionar un artefacto central para gobernar el acercamiento estratégico al esfuerzo de la prueba. Este define el acercamiento general que se empleará para probar el software y evaluar los resultados de esa comprobación, y es el artefacto que planeando los artefactos se referirá a los términos de dirección del trabajo detallado de comprobación.*
- Proporciona la visibilidad a los involucrados en el esfuerzo de la prueba que con la consideración adecuada se le ha dado a los aspectos de dirección del esfuerzo de la prueba, y donde se determina por los involucrados aprobar la estrategia.*

*Esta Estrategia de Prueba también apoya los objetivos específicos siguientes:*

- Identifica los artículos que deben ser los llevados a cabo a través de las pruebas.*
- Identifica la motivación para y las ideas detrás de las áreas de la prueba a ser cubierta.*
- Esboza el acercamiento de la comprobación que se usará.*
- Identifica los recursos requeridos y proporciona una estimación de los esfuerzos de la prueba.*
- Lista los elementos entregables del proyecto de prueba.*

#### 1.2 Alcance

*Define los tipos de pruebas como, Funcional, Usabilidad Fiabilidad, Funcionamiento, y Soportabilidad y si es necesario los niveles de prueba por ejemplo, Integración o Sistema que se dirigirán por esta Estrategia de la Prueba. También es importante proporcionar una indicación general de elementos significativos que se excluirán del alcance, sobre todo donde el público intencional podría asumir por otra parte razonablemente la inclusión de esos elementos.*

#### 1.3 Público intencional.

*Proporciona una descripción breve del público para quien usted está escribiendo a la Estrategia de la Prueba. Esto ayuda a los lectores de su documento a identificar si es un documento pensado para su uso, y ayuda a la prevención de usarse el documento impropiaamente.*

*Esta sección sólo debe ser aproximadamente tres a cinco párrafos en la longitud.*

## **1.4 Terminología del Documento y Acrónimos**

*Esta sección proporciona las definiciones de cualquier condición, las siglas, y abreviaturas requeridas propiamente para interpretar la Estrategia de la Prueba. Evita artículos de la inscripción que son en conjunto generalmente aplicable al proyecto y ese ya se define en el Glosario del proyecto. Incluye una referencia al Glosario del proyecto en la sección de las Referencias*

## **1.5 Referencias**

*Esta sección proporciona una lista de las referencias de los documentos en otra parte dentro de la Estrategia de la Prueba. Identifica cada documento por el título, versión (o número del informe si aplicable), fecha, y publicación de la organización o autor del original. Evita documentos de la inscripción que son influyentes pero no directamente referenciados. Especifica las fuentes de que pueden obtenerse las versiones oficiales de las referencias, como la nombres de intranet o códigos de referencia de documento. Esta información puede proporcionarse por la referencia a un apéndice o a otro documento.*

## **1.6 Estructura del documento**

*Esta subdivisión perfila lo que contiene el resto de la Estrategia de la Prueba y da una introducción de cómo el resto del documento está organizado. Esta sección puede eliminarse si se usa una Tabla de Volúmenes.*

## **2. Misión de Dirección de Evaluación**

*Proporciona una apreciación global de/las misión(es) que dirigirá la prueba detallada dentro de las iteraciones.*

### **2.1 Contexto del proyecto y Descripción General**

*Proporciona una descripción general breve del proyecto con la referencia específica o enfoque en las implicaciones importantes para el esfuerzo de la prueba. Incluye la información como el problema clave a resolver, los mayores beneficios de la solución, la arquitectura planeada de la solución, y una breve historia del proyecto. Apunta que dónde esta información se define suficientemente en otros documentos, usted podría incluir una referencia simplemente a esos otros documentos si es apropiado; sin embargo, puede salvar a los lectores del tiempo de estrategia de prueba y esfuerzo si una cantidad limitada de información se reproduce aquí: así que se debe usar al juicio de quien la use. Como una regla general, esta sección debe ser sólo aproximadamente tres a cinco párrafos en la longitud.*

### **2.2 Misión de Evaluación**

*Proporciona una breve declaración que define la(s) misión(es) para la prueba y esfuerzo de evaluación sobre el alcance del plan. Las pautas de la dirección de la misión podrían incorporar una o más preocupaciones incluyendo:*

- Encontrar tantos errores como sea posible
- Encontrar los problemas importantes, evaluar los riesgos de calidad percibidos
- Avistar sobre los riesgos del proyecto percibidos
- Certificar a una norma
- Verificar una especificación (los requisitos, diseño o demanda)
- Avizar sobre que la calidad del producto, satisfaga a los clientes
- Señalar sobre las pruebas
- Cumplir los mandatos trazados del proceso
- Cada misión proporciona un contexto diferente al esfuerzo de la prueba y cambia la manera en que la prueba deben aproximarse.

### 2.3 Motivación de la Prueba.

Proporciona una descripción de los elementos claves que motivarán el esfuerzo de la prueba en esta iteración. La prueba se motivará por muchas cosas de riesgos de calidad, los riesgos técnicos, los riesgos del proyecto, casos de uso, requisitos funcionales, requisitos no funcionales, elementos del diseño, fallos sospechosos o fallas, las demandas de cambio, y demás trazadas en el proyecto.

### 3. Dirección de Pruebas.

Es importante recordar que como una regla general es un acercamiento de la prueba apropiado es un contexto específico del proyecto individual. Como tal, los elementos definidos en la dirección diferirán de proyecto-a-proyecto dependiendo de la misión de la evaluación y los factores específicos de otros proyectos.

La Dirección de Prueba presenta una estrategia recomendada para analizar, diseñar, implementar y ejecutar las pruebas requeridas. Los Planes de la Prueba específicos identifican qué artículos serán los objetivos por probar y qué tipo de prueba se ejecutaran. Esta sección de la estrategia de la prueba describe cómo las pruebas serán llevadas a cabo..

Uno de los principales aspectos de la dirección de pruebas es la selección de las técnicas que se usarán. Esta estrategia de la prueba debe incluir un esbozo de cómo cada técnica puede diseñarse, implementada y ejecutada, y el criterio para saber que la técnica es útil y exitosa. Para cada técnica, proporciona una descripción de la misma y define por qué es una parte importante de la dirección de la prueba resumiendo brevemente cómo ayuda a lograr la(s) Misión(es) de Evaluación.

Según se defina cada aspecto del acercamiento, se debe considerar el impacto que tendrá sobre los recursos como el personal, herramientas y hardware de prueba y tomar nota del impacto.

#### 3.1 Medida y Alcance de la Prueba.

Describe qué estrategia se usará para medir el progreso del esfuerzo de la prueba. Al decidir en una medida de la estrategia, es importante considerar el consejo siguiente de Cem Kaner, 2000 "Los errores en las cuentas en las métrica, reflejan sólo una pequeña parte del trabajo y progreso del grupo de prueba. Muchas alternativas

*miran lo que tiene que ser hecho más estrechamente y lo que se ha hecho. Éstos serán a menudo más útiles y menos propenso estar al lado de los efectos del error de cuenta de la métrica".*

*Una buena estrategia de medida informará en las múltiples dimensiones. Se consideran las dimensiones siguientes, y seleccionan un subconjunto que es apropiado para su contexto del proyecto: los fondos (contra el producto y / o contra el plan), esfuerzo, resultados, los obstáculos, los riesgos (en la calidad del producto y / o probando la calidad), la tendencia histórica (por las iteraciones y / o por los proyectos).*

### **3.2 Identificar y Justificar las Pruebas.**

*Describe cómo se identificarán las pruebas y serán consideradas para la inclusión en el alcance del esfuerzo de la prueba cubierto por esta estrategia. Proporciona un listado de recursos que se usarán para estimular / dirigir la identificación y selección de pruebas específicas a ser ejecutadas, como los Catálogos de Prueba, los documentos de Requisitos, documentación del Usuario y / u Otras Fuentes de Referencia. Pueden encontrarse ejemplos de Catálogos de Prueba en los componentes del proceso enviados con RUP.*

### **3.3 Guía de Pruebas.**

#### **3.3.1 Técnica 1**

*Proporciona una breve introducción de más o menos un párrafo de la técnica seleccionada, mientras cubre las bases teóricas detrás de la técnica y la el riesgo de calidad general que este acarrea.*

Objetivo de la técnica:	<i>Explica el enfoque y meta de la técnica en relación con el riesgo de la calidad que trae consigo.</i>
Técnica:	<i>Esboza el procedimiento de alto nivel para la técnica, posiblemente como los pasos a llevar a término en la apreciación de los niveles generales.</i>
Responsable:	<i>Bosqueja una o más estrategias que pueden usarse con la técnica para observar los resultados de la prueba con precisión. El responsable combina elementos de ambos el método por el cual la observación puede hacerse y las características del resultado específico que indica éxito probable o fracaso. Con suerte, los responsables estarán ellos mismos verificando, permitiendo las pruebas automatizadas para hacer una valoración inicial de paso de la prueba o fracaso, sin embargo, se debe tener el cuidado para mitigar los riesgos inherentes en la determinación de los resultados automatizados.</i>
Herramientas Requeridas:	<i>Proporciona una lista simple de las herramientas específicas o un breve esbozo de cada tipo de herramientas que la técnica requiere.</i>

Objetivo de la técnica:	<i>Explica el enfoque y meta de la técnica en relación con el riesgo de la calidad que trae consigo.</i>
Criterio de Éxito:	<i>Explica cómo la técnica se juzgará como exitosa, dando criterio específico que se puede y se medirá.</i>
Consideraciones Especiales:	<i>Proporciona una lista de constreñimientos, dependencias u otras consideraciones que tendrán un impacto en la técnica, como habilidades del verificador o requisitos de recurso de prueba.</i>

### 3.3.2Técnica n+ 1

*Proporciona una breve introducción de más o menos un párrafo de la técnica seleccionada, mientras cubre las bases teóricas detrás de la técnica y la el riesgo de calidad general que este acarrea.*

Objetivo de la técnica:	<i>Explica el enfoque y meta de la técnica en relación con el riesgo de la calidad que trae consigo.</i>
Técnica:	<i>Esboza el procedimiento de alto nivel para la técnica, posiblemente como los pasos a llevar a término en la apreciación de los niveles generales.</i>
Responsable:	<i>Bosqueja una o más estrategias que pueden usarse con la técnica para observar los resultados de la prueba con precisión. El responsable combina elementos de ambos el método por el cual la observación puede hacerse y las características del resultado específico que indica éxito probable o fracaso. Con suerte, los responsables estarán ellos mismos verificando, permitiendo las pruebas automatizadas para hacer una valoración inicial de paso de la prueba o fracaso, sin embargo, se debe tener el cuidado para mitigar los riesgos inherentes en la determinación de los resultados automatizados.</i>
Herramientas Requeridas:	<i>Proporciona una lista simple de las herramientas específicas o un breve esbozo de cada tipo de herramientas que la técnica requiere.</i>
Criterio de Éxito:	<i>Explica cómo la técnica se juzgará como exitosa, dando criterio específico que se puede y se medirá.</i>
Consideraciones Especiales:	<i>Proporciona una lista de constreñimientos, dependencias u otras consideraciones que tendrán un impacto en la técnica, como habilidades del verificador o requisitos de recurso de prueba.</i>

#### 4. Necesidades de Ambiente.

*Esta sección presenta los recursos no humanos requeridos para la Estrategia de la Prueba.*

##### 4.1 Hardware base del sistema.

*La Tabla siguiente establece los recursos del sistema para el esfuerzo de la prueba presentado en esta Estrategia de la Prueba.*

*No pueden entenderse los elementos específicos del sistema de la prueba totalmente en las iteraciones tempranas, así que se debe esperar completar esta sección con el tiempo. Se recomienda que el sistema simule el ambiente de la producción, mientras se reduce el acceso coexistente y tamaño de la base de datos*

*Se pueden agregar o anular los artículos que se consideren.*

Recursos del Sistema		
Recurso	Cantidad	Nombre y Tipo



#### 4.2 Elementos base software en la Prueba de Ambiente.

Los siguientes elementos de software se requieren en el ambiente de la prueba para esta Estrategia de la Prueba.

Se pueden agregar o anular los artículos que se consideren.

Nombre del elemento de Software	Versión	Tipo y otras Anotaciones

#### 4.3 Productividad y Herramientas de Soporte.

Las siguientes herramientas se emplearan para darle soporte al proceso de prueba para la estrategia actual. Se pueden agregar o anular los artículos que se consideren.

Categoría de la Herramienta y Tipo	Nombre de la herramienta.	Privativa o Liberada	Versión

#### 4.4 Configuración del ambiente de Prueba.

Las Configuraciones de Ambiente de Prueba siguientes se proporcionan y apoyan este proyecto.

Nombre de la configuración	Descripción	Implementación de la configuración física

Nombre de la configuración	Descripción	Implementación de la configuración física

## 5. Responsabilidades, Personal y Necesidades de Entrenamiento.

*Esta sección presenta los recursos requeridos para dirigir el esfuerzo de la prueba enfocado en la Estrategia de Prueba las responsabilidades principales, y el conocimiento o habilidades adquiridas que requirieron de esos recursos*

### 5.1 Personal y Roles

*Esta tabla muestra el aseguramiento del personal para el esfuerzo de la prueba.*

*Se pueden agregar o anular los artículos que se consideren.*

Recursos Humanos		
Rol	Mínimos Recursos Recomendados (numero de rol a tiempo completo asignado)	Responsabilidades Especificas Y Comentarios

Recursos Humanos		
Rol	Mínimos Recursos Recomendados (numero de rol a tiempo completo asignado)	Responsabilidades Especificas Y Comentarios
Administrador de Pruebas		<p><i>Realiza el control de dirección.</i></p> <p><i>Las responsabilidades incluyen:</i></p> <ul style="list-style-type: none"> <li>• <i>planeamiento y logísticas</i></li> <li>• <i>acuerdos de la misión</i></li> <li>• <i>identificas las motivaciones</i></li> <li>• <i>adquirir los recursos apropiados</i></li> <li>• <i>dirigir los reportes presentes</i></li> <li>• <i>abogar por los intereses de prueba</i></li> <li>• <i>evaluar efectividad de esfuerzo de la prueba</i></li> </ul>
Analista de Prueba		<p><i>Identificar y definir las pruebas especificas a realizar.</i></p> <p><i>Las responsabilidades incluyen:</i></p> <ul style="list-style-type: none"> <li>• <i>identificar las ideas de la prueba</i></li> <li>• <i>definir los detalles de la prueba</i></li> <li>• <i>determinar los resultados de la prueba</i></li> <li>• <i>demandar cambios en el documento</i></li> <li>• <i>evaluar la calidad del producto</i></li> </ul>
Diseñador de Prueba		<p><i>Define el acercamiento técnico a la aplicación del esfuerzo de la prueba.</i></p> <p><i>Las responsabilidades incluyen:</i></p> <ul style="list-style-type: none"> <li>• <i>definir el acercamiento de la prueba</i></li> <li>• <i>definir la arquitectura de automatización de prueba</i></li> <li>• <i>verificar las técnicas de la prueba</i></li> <li>• <i>definir los elementos a probar.</i></li> <li>• <i>estructurar la prueba de la aplicación</i></li> </ul>

Recursos Humanos		
Rol	Mínimos Recursos Recomendados (numero de rol a tiempo completo asignado)	Responsabilidades Especificas Y Comentarios
Probador		<p><i>Implementación y ejecución de las pruebas.</i></p> <p><i>Las responsabilidades incluyen:</i></p> <ul style="list-style-type: none"> <li>• <i>implementación de pruebas y conjunto de pruebas</i></li> <li>• <i>ejecución de los conjuntos de la prueba</i></li> <li>• <i>registrar los resultados</i></li> <li>• <i>analizar y recuperar los fracasos de la prueba</i></li> <li>• <i>documentar las incidencias</i></li> </ul>
Administrador de Prueba de Sistema.		<p><i>Asegurar el ambiente de la prueba y administrar y mantienen los recursos.</i></p> <p><i>Las responsabilidades incluyen:</i></p> <ul style="list-style-type: none"> <li>• <i>administrar el sistema de dirección de prueba</i></li> <li>• <i>instalar y apoyar el acceso a, y recuperación de, configuraciones de ambiente de prueba</i></li> <li>• <i>apoyar la administración de datos de la prueba y base de la prueba (la base de datos).</i></li> </ul>
Administrador de Base de Datos.		<p><i>Asegurar los datos de la prueba (la base de datos) que se manejan en el ambiente y recursos y se mantienen.</i></p> <p><i>Las responsabilidades incluyen:</i></p> <ul style="list-style-type: none"> <li>• <i>apoyar la administración de datos de la prueba y base de la prueba (la base de datos).</i></li> </ul>

Recursos Humanos		
Rol	Mínimos Recursos Recomendados (numero de rol a tiempo completo asignado)	Responsabilidades Especificas Y Comentarios
Diseñador		<p><i>Identificar y define las operaciones, atributos, y asociaciones de las clases de la prueba.</i></p> <p><i>Las responsabilidades incluyen:</i></p> <ul style="list-style-type: none"> <li><i>definir las clases de la prueba exigidas para apoyar los requisitos de la prueba definidos por el equipo de la prueba</i></li> </ul>
Implementador		<p><i>Implementar la unidad prueba y clasificar la misma en paquetes de la prueba.</i></p> <p><i>Las responsabilidades incluyen:</i></p> <ul style="list-style-type: none"> <li><i>crear los componentes de la prueba exigidos apoyar los requisitos de la prueba definidos por el diseñador</i></li> </ul>

## 5.2 Aseguramiento del personal y necesidades de entrenamiento.

*Esta sección esboza cómo asegurar el personal y entrenar; la necesidades de los roles la prueba para el proyecto.*

*La manera de asegurar y entrenar el personal variará de a proyecto. En caso de esta sección ser parte de una Estrategia de la Prueba, se debe indicar en qué puntos en el ciclo de vida del proyecto se necesitan las diferentes habilidades y número de personal. En los Planes de Prueba de Iteración, se debe enfocar principalmente dónde y en que se debe entrenar durante la Iteración.*

*Pensar el entrenamiento, necesidades, y horario se debe basar Justo a Tiempo (JIT) hay a menudo una tendencia para asistir demasiado lejos de antemano al entrenamiento de su uso cuando el equipo de la prueba tiene debilidades claras. Haciendo esto introduce el riesgo de ser olvidado el entrenamiento cuando se necesita.*

*Mirar las oportunidades de combinar la compra de herramientas de productividad con entrenar en esas herramientas, coloca al vendedor a demorarla entrega del entrenamiento simplemente hasta antes de que usted lo necesite. Si usted cuenta con suficiente personal, considere el entrenamiento entregado de una manera personalizada para usted, posiblemente a su propio sitio.*

*El equipo de la prueba requiere a menudo al apoyo y habilidades de otros miembros del equipo. Asegúrese que usted coloca en su estrategia la disponibilidad apropiada de personal de apoyo: Administradores del sistema, Administradores de la Base de datos, y Diseñadores que son exigidos para habilitar el esfuerzo de la prueba.*

## 6. Riesgos, Dependencias, Retos y Limitaciones.

*Lista cualquier riesgo que puede afectar la ejecución exitosa de esta Estrategia de Prueba, e identifica la mitigación y estrategias de contingencia para cada riesgo. También indica una clasificación jerárquica para ambos la probabilidad de ocurrencia y el impacto si el riesgo ocurre.*

Riesgo	Estrategia de Mitigación	Contingencia (Si ocurre el riesgo)

*Lista cualquier dependencia identificada durante el desarrollo de esta Estrategia de Prueba que puede afectar su ejecución exitosa si esas dependencias no se respetan. Típicamente estas dependencias relacionan a las actividades en la ruta crítica que son requisitos previos o post-requisitos a actividades precedentes (o subsecuente). Se debe considerar en las responsabilidades confiando en otros equipos o miembros del personal externo al esfuerzo para completar prueba, estimando el tiempo en dependencia de otras tareas planeadas.*

Dependencias	Impacto Potencial de la Dependencia	Ejecutor

*Lista cualquier reto asumido durante el desarrollo de esta Estrategia de Prueba que puede afectar su ejecución exitosa si la meta trazada es probada incorrecta. Las metas podrían llevar a relacionar el trabajo con lo que otros equipos están haciendo, ya que ciertos aspectos del producto o ambiente son estables.*

Meta a ser probada	Impacto de la Meta si es incorrecta	Ejecutor

*Lista cualquier limitación que ha tenido un efecto negativo en el esfuerzo de la prueba, o en la manera en que se ha aproximado la Estrategia de Prueba.*

Limitación	Impacto de la Limitación en el Esfuerzo de Prueba	Ejecutor

## 7. Administración de Procesos and Procedimientos.

*Un bosquejo de los procesos y procedimientos que serán usados cuando los problemas se levantan con la Estrategia de Prueba y su promulgación.*

### 7.1 Reporte de Problema, Escalado, y Emitir Resolución

*Define cómo se procesaran los problemas que se reportaran y escalaran, y el proceso a ser seguido para lograr la resolución.*

### 7.2 Estrategia de Trazabilidad.

*Considera las estrategias apropiadas de Trazabilidad para:*

- *Proteger la Prueba contra las Especificaciones, permite la medir la extensión la prueba.*
- *Motivaciones de Prueba - permite a la valoración de la relevancia de la prueba para ayudar a determinar si se mantiene o retira la prueba.*
- *Elementos de Plan de software - habilita el rastreado de cambios en el plan subsecuentes que harían necesario recomenzar las pruebas o retirarlas.*
- *Demandas de Cambio resultantes - habilita las pruebas que determinaron necesidad de cambio para ser*

*identificadas y re-comenzadas para verificar que la demanda del cambio se ha completado con éxito*

### **7.3 Aprobación y Sellado (Firma del documento)**

*Esboza la el proceso de aprobación, y lista los titulares del trabajo que inicialmente deben aprobar la estrategia, así como la firma de una ejecución satisfactoria de la misma.*