

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
Facultad 9



Título: Guía para la utilización de los Casos de Uso en las
Pruebas de Aceptación.

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

AUTORES:

Liusdel Castro Berriel

Niurka Cruz Benítez

TUTORA:

MSc. Yeleny Zulueta Véliz

Ciudad de La Habana, Julio, 2008

"Año 50 de la Revolución"

FRASE

*"El hombre no se da cuenta de cuanto puede hacer, más
que cuando realiza intentos, medita y desea"*

"Ugo Foscolo"

DEDICATORIA

DEDICATORIA

A mi madre y mi abuela por apoyarme en todo y por la confianza depositada en mí.

A mi hermano por su ayuda en todos estos años.

A mi novia por guiarme siempre y por todo su cariño.

A todos mis amigos que me han ayudado de una forma u otra.

Y a mí por todo el esfuerzo dedicado.

Lúsdel Castro Berriel.

Este trabajo es dedicado a mi Padre que estuvo ayudándome en todo tiempo, esta tesis es su obra a través de mí, su sabiduría, su dedicación estuvo siempre conmigo; porque él fue quien permitió que estudiara en esta Universidad. Lo dedico especialmente a mis padres Ada Benítez Rossette y Ángel Cruz Romero por ser ellos lo más importante para mí en este mundo y este trabajo es el regalo que puedo darles por todo su esfuerzo, su ayuda incondicional, su amor, por todos los años de estudiante que me han guiado hasta obtener este título. Dedico este trabajo a mis abuelos, a todos mis familiares por ser tan especiales conmigo. A mis mejores amigas: Aimara y Caridad y mi hija de cariño Roxana y todos amigos y compañeros de la universidad que siempre estaban a mi lado en los momentos tristes y alegres de la vida. A esta universidad por todas los recursos y beneficios que puso a mi disposición.
Núrká Cruz Benítez.

AGRADECIMIENTOS

AGRADECIMIENTOS

Quiero agradecer a todas las personas que hicieron posible el desarrollo de este trabajo y a todos aquellos que me han apoyado a lo largo de la carrera, pero en especial:

A mi abuela por brindarme todo su cariño y apoyarme en todo momento.

A mi madre por haberme guiado en todos estos años, aún en decisiones difíciles.

A mi hermano por ser parte de mi vida.

A mi novia que tantos días dedicó conmigo para desarrollar este trabajo.

A mi tutora Yeleny Zulueta que me ayudó en todo y por brindarme todo su conocimiento para lograr este resultado.

A todos aquellos que se preocuparon por mi trabajo.

A los que me apoyaron en los momentos difíciles.

A Fidel y La Revolución por darme la posibilidad de realizar mi sueño y convertirme en un buen profesional.

A todos muchas gracias...

Líusdel Castro Berriel.

AGRADECIMIENTOS

Agradezco a:

Mi Mejor Amigo a Jesús porque sin El no estuviera en esta Universidad, que en todos los momentos ha estado conmigo y por darme todo su amor y animarme en todo tiempo.

A Mis Padres amados Ángel Cruz Romero y Ada Benítez Rossette que han estado siempre dándome palabras de exhortación y consuelo, apoyándome en todas mis decisiones siendo mi refugio en los tiempos difíciles.

A mi tío Armando Rossette por ser ese padre aquí en la Habana, que por tantos días y noches se sacrificó por mí.

A mis amistades incomparables Caridad Ortiz, Aímara Bárbara y Arisbel Luna compañero de estudios incansables y a todas mis amistades y hermanos que me han apoyado y ayudado en todos los días de estudiante.

A la profesora Lili Martínez amiga y mi mami en la UCI y su familia que en los últimos años de la Universidad han estado conmigo ayudándome y qué sería de mí sin su apoyo incondicional y por darme fuerza en los momentos más difíciles y por el tiempo que dedique a la elaboración de la tesis en su apto.

A nuestra tutora Yeleny, por su inestimable cooperación, su preocupación y ayuda, por la paciencia y esmero, por mantenerse siempre optimistas y formar en nosotros mejores valores humanos.

A Sandý miembro del tribunal y los demás integrantes y a todos los profesores que de una forma u otra han aportado un granito de arena para lograr la confección de mi trabajo de diploma.

Níurka Cruz Benítez.

RESUMEN

RESUMEN

En el mundo todas las empresas constructoras de software admiten que enfrentan problemas con el rechazo de los productos por parte del cliente, esto se debe a la mala calidad del proceso de pruebas, por lo que se necesita buscar una solución para resolver esta situación.

El presente trabajo está enfocado a realizar una guía para el desarrollo de las pruebas de aceptación apoyándose en los casos de uso, artefacto sumamente importante dentro de este proceso. Estas pruebas no tienen otro objetivo que validar el producto para ver si lo que se hace es lo que realmente se espera. Las pruebas de aceptación son realizadas por los clientes quienes son lo encargados de encontrar todas las no conformidades existentes y solicitar todos los cambios necesarios.

El aporte de este trabajo tiene un gran impacto en la situación del software en la Universidad de Las Ciencias Informáticas (UCI), ya que anteriormente no existía ninguna guía para realizar las pruebas de aceptación. Con la solución del trabajo se espera lograr que los equipos de desarrollo entreguen un producto verdaderamente exitoso y capaz de satisfacer al cliente.

Como objetivo principal se tiene precisamente el desarrollo de una guía para la utilización de los Casos de Uso en las Pruebas de Aceptación, que puede ser aplicada en todos los proyectos de calidad de la UCI para lograr mejor calidad en el desarrollo del software. Para darle cumplimiento a este objetivo principal se plantearon otros más específicos:

- Analizar del marco existente que se sigue para la realización de los Casos de Uso.
- Elaborar una descripción de las principales actividades a seguir a la hora de hacer los Casos de Uso de Prueba.
- Validar la solución de la guía.

Gracias a este trabajo la UCI puede contar con una guía para realizar las pruebas de aceptación con la mayor calidad posible y poseer mejor organización, experiencia, madurez, rapidez, habilidad y control de las pruebas en el proyecto de calidad.

PALABRAS CLAVES

Calidad, Pruebas, Pruebas de Aceptación, Casos de Uso, Guía.

CONTENIDO

CONTENIDO

DEDICATORIA	I
AGRADECIMIENTOS	II
RESUMEN	VI
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	6
1.1- Introducción.....	6
1.2- Conceptos Asociados al Dominio del Problema.....	6
1.2.1- Calidad de Software.....	6
1.2.2- Pruebas del Software.....	7
1.2.3- Pruebas de Aceptación.....	7
1.3- Gestión de la Calidad del Software.....	8
1.3.1- Planificación de la Calidad del Software.....	9
1.3.2- Control de la Calidad del Software.....	12
1.3.3- Aseguramiento de la Calidad del Software.....	12
1.3.4- Mejora de La Calidad del Software.....	16
1.4- Gestión de Pruebas.....	19
1.4.1- Pruebas.....	19
1.4.2- Objetivos de las pruebas.....	22
1.4.3- Características de las pruebas.....	23
1.4.4- Principios de las pruebas.....	24
1.5- Pruebas de Aceptación.....	25
1.5.1- Definición de las pruebas de aceptación.....	28
1.5.2- Características de las pruebas de aceptación.....	29
1.6- Flujo de trabajo para pruebas definidos por Pressman.....	30
1.7- Conclusiones Parciales.....	32
CAPÍTULO 2: ANÁLISIS DE OTRAS SOLUCIONES	33
2.1- Introducción.....	33
2.2- Proceso de calidad en la UCI.....	33
2.3- Estado de las pruebas en el mundo.....	35

CONTENIDO

2.4- El proceso de pruebas en Cuba.	36
2.4.1- Estado de las pruebas de aceptación.	37
2.5- Descripción actual del dominio del problema.	38
2.6- Situación Problemática.	40
2.7- Flujo de trabajo en la UCI para el desarrollo de la pruebas.	41
2.8- Entrevistas realizadas.	42
2.8.1- Análisis de la entrevista a Ramsés Delgado.	42
2.8.2- Análisis de la entrevista a Dialexis Acosta Medina.	43
2.9- Investigaciones desarrolladas en la UCI acerca de pruebas de aceptación.	44
2.10- Conclusiones Parciales.	46
CAPÍTULO 3. PROPUESTA DE SOLUCIÓN PARA EL DESARROLLO DE LAS PRUEBAS DE ACEPTACIÓN.	47
3.1- Introducción.	47
3.2- Pasos para un buen proceso de pruebas.	47
3.3- Descripción del Flujo de Trabajo Final que le da solución a la investigación.	50
3.3.1- Actividades fundamentales.	51
3.4- Roles que intervienen en el proceso.	53
3.4.1- Descripción de los roles.	54
3.5- Artefactos presentes en la solución.	58
3.5.1- Artefactos más importantes.	60
3.6- Diagrama de actividades.	63
3.7- Opiniones de expertos.	64
3.8- Conclusiones Parciales.	64
CONCLUSIONES	65
RECOMENDACIONES	66
REFERENCIAS BIBLIOGRÁFICAS	67
BIBLIOGRAFÍA	68
ANEXOS	70
GLOSARIO DE TÉRMINOS	82

INTRODUCCIÓN

INTRODUCCIÓN

Durante el desarrollo de la informática a nivel mundial uno de los problemas más notable era la calidad del software. Desde la década del 70, este tema ha sido motivo de preocupación para especialistas, ingenieros, investigadores y comercializadores de software, los cuales han realizado gran cantidad de investigaciones a partir del planteamiento de dos grandes interrogantes: ¿Cómo obtener un software con calidad? y ¿Cómo evaluar la calidad del software?, definitivamente estas interrogantes han hecho posible la creación de normas con tal de que organizaciones destinadas al desarrollo del software se apoyen en ellas y realicen un mejor trabajo. La calidad es subjetiva y cada una de las compañías productoras de software tiene diversas necesidades y criterios para valorar de diferentes formas los atributos de un producto o servicio. La obtención de un software con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software, que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software. El software se ha convertido en un tema crítico en la sociedad moderna mundial. Todos parecen necesitar mejores software en el menor tiempo posible y a menor costo. Los métodos intuitivos de desarrollo de software que se usan actualmente son, básicamente, aquellos que los propios individuos "artesanalmente" siguen, los cuales sólo servirán mientras la sociedad pueda tolerar la falta de predicción que ellos acarrearán, es por ello que se hace cada vez más imponente el tema de la calidad de los productos de software que se producen en todo el mundo.

La calidad es un aspecto que ha adquirido mucho interés por parte de todas las empresas de software en el mundo debido al alto grado de exigencia que presentan clientes respecto al producto, especialmente en el desarrollo de software. El interés por la misma crece de forma continua, a medida que los clientes se vuelven más selectivos y comienzan a rechazar productos poco fiables o que realmente no dan respuesta a sus necesidades. La entrega de los productos con calidad le aporta al país y a la institución un gran prestigio, eleva el desarrollo en el campo tecnológico, y aumenta la fiabilidad en el trabajo realizado.

Cuando se va a definir la calidad del software se pueden adoptar diferentes aproximaciones. Como primera aproximación está la importancia de saber diferenciar entre la eficiencia del Producto software y del Proceso de desarrollo. No obstante, las metas que se establezcan para la calidad del producto van a determinar las metas a establecer para el proceso de desarrollo, ya que la calidad del producto

INTRODUCCIÓN

va a estar en función del proceso de desarrollo. Sin un buen proceso de desarrollo es casi imposible obtener un buen producto (Quesada, 2007).

Para demostrar si un producto tiene la calidad requerida o no, se desarrolla el proceso de pruebas que es el encargado de detectar fallos (Quesada, 2007). En el mundo de las pruebas de software con el objetivo de medir la calidad de todo producto, existen varios niveles de pruebas. Se encuentran las Pruebas de Unidad que consisten en la prueba de cada módulo de forma aislada del resto del sistema, Pruebas de Integración que son las que se realizan a medida que los diferentes módulos del sistema se integran en el mismo, Pruebas de Sistema que se realizan cuando se han integrado todos los módulos, y tiene como objetivo principal comprobar que el sistema satisface los requisitos del usuario y Pruebas de Aceptación que son las que se realizan una vez que el sistema se ha implantado en su entorno real de funcionamiento (Antonio, 1999).

Son las pruebas de aceptación la idea central sobre la que gira la investigación, las cuales son muy importantes a la hora de validar y aceptar el producto final.

En Cuba se encuentra la Universidad de las Ciencias Informáticas (UCI), donde se desarrollan software y todos sus productos se hacen con la mayor calidad posible y siempre pensando en la opinión del cliente.

En la UCI como se mencionó antes, la calidad se tiene muy en cuenta y para lograr un producto con la mejor aceptación del cliente se realiza el proceso de pruebas que es tan importante en el desarrollo del software. Las pruebas son realizadas por grupos de calidad, encargados de encontrar todos los errores que contiene el software, estos grupos representan proyectos que se encuentran en cada una de facultades y su principal objetivo es llevar a cabo el proceso de pruebas, sin importar el tipo de pruebas que sea.

Para realizar las pruebas fundamentalmente los jefes de proyectos de calidad se basaban en aspectos como requerimientos y casos de uso que son muy importantes en esta tarea y que ayudan a obtener una visión.

Después de llevar a cabo un análisis de cómo se estaban desarrollando las pruebas en la UCI, no se obtuvieron los resultados esperados ya que existe una gran necesidad de obtener productos con calidad y en la Universidad no se realizan este tipo de pruebas de Aceptación al cliente, solo se realiza una revisión final, porque no se utilizan correctamente los Casos de Uso, debido a la inexistencia de una guía que permita formalizar una manera adecuada de desarrollarlas.

INTRODUCCIÓN

La Universidad se dio la tarea de realizar un estudio profundo para lograr que el proyecto Calidad tuviera una guía para una mejor realización de las Pruebas de Aceptación de un producto con la utilización de los Casos de Uso.

Después de analizar la situación problemática de la investigación, al **problema científico** que se le dará solución es ¿Cómo utilizar los Casos de Uso en la concepción y realización de las Pruebas de Aceptación de un producto?

El **objeto de estudio** de esta investigación es el Proceso de Pruebas de Aceptación en el Desarrollo del Software y el **objetivo general** que se quiere lograr es establecer una guía que permita realizar las Pruebas de Aceptación de un producto utilizando la documentación relacionada con los Casos de Uso.

El **campo de acción** de la investigación son: Los Casos de Uso en el Proceso de Pruebas de Aceptación en el Desarrollo del Software.

Al finalizar la investigación se esperan los **resultados** siguientes:

- Análisis de los marcos existentes en el área.
- Descripción de las actividades fundamentales para la utilización de los Casos de Uso en las Pruebas de Aceptación.
- Conjunto de Principios que fundamenten una guía para una mejor aplicación de los casos de uso de Pruebas.

En la investigación se obtuvo como **objetivos específicos**:

- Analizar del marco existente que se sigue para la realización de los Casos de Uso.
- Elaborar una descripción de las principales actividades a seguir a la hora de hacer los Casos de Uso de Prueba.
- Validar la solución de la guía.

Después de analizar todo lo anterior se elaboró la **hipótesis de la investigación** que plantea que si se desarrolla un profundo análisis y estudio de los casos de uso en el Proceso de Pruebas de un Software se logrará definir una guía que permita realizar las Pruebas de Aceptación usando esta documentación.

INTRODUCCIÓN

La **variable independiente** de la investigación es el análisis y estudio de los casos de uso en el Proceso de Pruebas de un Software y de ella **depende** la definición de la guía que permita realizar las Pruebas de Aceptación.

Para realizar la investigación y obtener los resultados esperados se deben cumplir estas **tareas**:

- Analizar las actividades involucradas en el proceso de pruebas de aceptación.
- Conformar la fundamentación teórica de la investigación.
- Definir el entorno metodológico para la utilización de los Casos de Uso en las Pruebas de Aceptación.
- Definir elementos para la evaluación de la propuesta.
- Ejecutar evaluación con método de multicriterio de expertos.
- Revisar el documento final.

Para el desarrollo de la investigación se aplican diferentes **métodos científicos**, tanto teóricos como empíricos.

En los teóricos se escogió el **analítico-sintético**, el cual sirve como método para buscar la esencia de las pruebas de aceptación, identificar las principales actividades llevadas a cabo en el proceso de validación en los proyectos de calidad en la Universidad de las Ciencias Informáticas. Para después realizar una síntesis de todos los elementos más importantes dentro de este proceso y tratar de llegar a la elaboración de una guía para la realización de pruebas de aceptación del cliente.

El método **histórico-lógico** se utiliza para analizar la trayectoria por todos los elementos tratados en el marco teórico de la investigación, también se manejan las principales etapas del proceso de pruebas y conocer cual ha sido el resultado de ellos en la universidad en el control de la calidad, para después mediante el estudio de la teoría y la práctica llegar a deducciones a partir de los conocimientos poseídos acerca del tema principal de la investigación. En este método se utiliza con frecuencia la búsqueda y consulta bibliográfica.

No podrían faltar los métodos empíricos, **la observación** se utiliza en la atención a la evolución de los procesos de validación en los proyectos de la universidad pudiendo realizar un análisis del comportamiento de los mismos y poder utilizar experiencias exitosas para elaborar una mejor y basada principalmente en errores anteriores.

Finalmente se utilizará **la entrevista** para la recopilación de toda información necesaria acerca del proceso de pruebas de aceptación que sea importante para la investigación.

INTRODUCCIÓN

Para la realización de la entrevista se escogió como población a los 11 líderes de los proyectos de calidad que están en las Facultades de la UCI y se aplicó un muestreo probabilístico y dentro de este el aleatorio simple, donde se obtuvo como muestra tres líderes de proyecto, el de la Facultad 9, la Facultad 3 y Facultad 7.

La tesis consta de 3 Capítulos:

Capítulo 1: Fundamentación teórica. Donde se expone un basamento teórico y conceptual acerca de la situación actual de la calidad de software, además de las características fundamentales de las pruebas que se llevan a cabo para garantizar un exitoso despliegue del sistema informático que se vaya a implantar.

Capítulo 2: Análisis de otros resultados. En este capítulo se profundiza acerca del estado actual de las pruebas de aceptación en la UCI y se lleva a cabo un estudio de trabajos existentes relacionados con el tema de la investigación.

Capítulo 3: Propuesta de Solución. En este capítulo se expone de manera clara la propuesta de solución con su correspondiente flujo de trabajos, así como los artefactos y roles que intervienen.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

1.1- Introducción.

Una teoría es una colección de conceptos sobre algún área del mundo real que interesa o preocupa, que facilita su explicación, predicción o intervención. Con teorías se explica cómo y porqué las cosas ocurren como ocurren. Se precede lo que ocurrirá dadas cómo son las cosas. Y escoger formas de actuar para que las cosas salgan del modo que se desea.

Es precisamente la fundamentación teórica lo que trata este capítulo, pues constituye el punto de partida para realizar la investigación. Aborda todo lo relacionado con la calidad del software y principalmente con el desarrollo de las pruebas y todo lo que interviene en este proceso que es tan importante en el desarrollo del software, de aquí que se haya hecho necesario plantear fundamentos teóricos para estudiar el origen y evolución de las mismas.

1.2- Conceptos Asociados al Dominio del Problema.

Este epígrafe les muestra los conceptos más importantes y que están relacionados con el problema, para de esta manera lograr un mejor entendimiento del capítulo, entre ellos se encuentran: Calidad de Software, Pruebas del Software y Pruebas de Aceptación.

1.2.1- Calidad de Software.

La calidad de software es un aspecto que ha evolucionado mucho debido al incremento de la necesidad por la calidad y la eficiencia en el software.

Su trascendencia es tal que la calidad ya no es una ventaja competitiva, sino una condición necesaria que deben cumplir las empresas de desarrollo para sobrevivir en el mercado.

(Kan, 2003).

Este no es el único concepto asociado, pues muchos más aportaron su punto de vista.

Capacidad del Software para satisfacer las necesidades de los usuarios clasificando diferentes aspectos en categorías conocidas como parámetros". (Juran, 1970).

"Conformidad o cumplimiento de los requisitos". (Crosby, 1979).

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

“Obtención de un grado elevado de satisfacción de los usuarios”. (Humphrey, 2001).

Recientemente la ISO la ha definido como:

“El grado de cumplimiento de una serie de características que satisfacen determinados requisitos”. (ISO 9001:00, 2000).

La calidad de un software abarca todo el proceso de desarrollo del software, porque cuando se habla de calidad no solo se refiere a la calidad del producto sino también a la calidad de los procesos, el cual es determinante, porque de cierto modo garantiza o reduce enormemente la probabilidad de entregar un software defectuoso o con muy baja calidad.

1.2.2- Pruebas del Software.

Las pruebas de software son el proceso de ejecución de un programa con la intención de descubrir un error, el objetivo fundamental es diseñar pruebas que sistemáticamente saquen a la luz diferentes clases de errores haciéndolo con la menor cantidad de tiempo y esfuerzos. (Pressman, 2002).

Las pruebas del software se pueden traducir como una revisión del sistema con el objetivo de encontrar problemas antes que estos sean encontrados por el cliente, el éxito de las pruebas puede mejorar la percepción de la calidad del usuario final, evitando que un defecto grave se entregue juntamente con el sistema. La prueba del software es un mecanismo para evaluar la calidad de acuerdo a la funcionalidad, facilidad de uso, fiabilidad y desempeño.

1.2.3- Pruebas de Aceptación.

Permite determinar si el software cumple con los requisitos previstos para su uso. Define las actividades a realizar por el adquiriente, el proveedor o una tercera parte independiente, para validar si el uso de los productos o servicios del proyecto satisface a los adquirientes. Las actividades de este proceso son: (ISO/IEC 9000:2000).

- Implementación del proceso.
- Validación.

El uso de cualquier producto de software tiene que estar justificado por las ventajas que ofrece. Sin embargo, antes de empezar a usarlo es muy difícil determinar si sus ventajas realmente justifican su uso. El mejor instrumento para esta determinación es la llamada 'prueba de aceptación'. En esta

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

prueba se evalúa el grado de calidad del software con relación a todos los aspectos relevantes para que el uso del producto se justifique.

Para eliminar la influencia de conflictos de intereses, y para que sea lo más objetiva posible, la prueba de aceptación nunca debería ser responsabilidad de los ingenieros de software que han desarrollado el producto.

Para la preparación, la ejecución y la evaluación de la prueba de aceptación ni siquiera hacen falta conocimientos informáticos. Sin embargo, un conocimiento amplio de métodos y técnicas de prueba y de la gestión de la calidad en general facilita esta labor.

La persona adecuada (o el equipo adecuado) para llevar a cabo la prueba de aceptación dispone de estos conocimientos y además es capaz de interpretar los requerimientos especificados por los futuros usuarios del sistema de software en cuestión.

Las pruebas de aceptación son realizadas con el fin de que el usuario o el cliente puedan dar por bueno el sistema, verifican si los requerimientos de los usuarios han sido satisfechos y para que no exista ni una función que no haya sido revisada y corregida. (Pressman, 2002).

1.3- Gestión de la Calidad del Software.

La Gestión de la Calidad de Software (CS) se basa en la determinación y aplicación de las políticas. Es una actividad esencial en cualquier empresa de software para asegurar la calidad de sus productos, y la competitividad frente a la oferta del mercado. Es un conjunto de actividades de la función general de la dirección que determina la calidad, los objetivos y las responsabilidades. (ISO 9000:2000).

La Gestión o Administración de la Calidad se aplica normalmente a nivel de empresa. También puede haber una gestión de la calidad dentro de la gestión de cada proyecto. El propósito de la Administración de la CS es, en primer lugar, entender las expectativas del cliente en términos de calidad, y poner en práctica un plan proactivo para satisfacer esas expectativas. Dado que la calidad está definida por el cliente, podría parecer que es completamente subjetiva. De cualquier forma, hay muchas cosas acerca de la calidad que pueden hacerse objetivamente. Esto requiere examinar cada una de las características individuales del software y determinar una o más métricas que pueden recolectarse para reflejar dichas características. Por ejemplo, una característica de calidad puede ser

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

que la solución tenga la menor cantidad de errores. Esta característica puede medirse contando los errores y defectos de la solución.

La Administración de la Calidad no es un evento, en un proceso y una forma de pensamiento. Un producto de software consistente, de alta calidad no puede producirse a partir de un proceso malo. Existe la necesidad de un ciclo constante de medir la calidad, actualizar el proceso, medir otra vez, actualizar, etc.

Para hacer que la administración de calidad del software funcione, es vital recolectar métricas. Si no se capturan métricas será difícil mejorar los procesos a partir de una iniciativa de administración de calidad. Uno de los propósitos de la administración de la calidad del software es encontrar errores y defectos en el proyecto tan pronto como sea posible. Entonces, un buen proceso de administración de calidad tomará más esfuerzo y costo. De cualquier manera, habrá una gran recompensa al tiempo que el proyecto avanza.

Por ejemplo, es mucho más fácil arreglar un problema con los requerimientos de negocio durante la fase de análisis que tener que arreglar problemas durante las pruebas. En otras palabras, el equipo de proyecto debe intentar mantener una alta calidad durante el proceso de desarrollo de los productos de software, en vez de esperar arreglar problemas durante las pruebas cercanas al final del proyecto (o en el peor de los casos, cuando el cliente encuentra el problema después que el proyecto se completó).

Desde el punto de vista de la calidad, la Gestión de la Calidad del Software está formada por 4 partes, las cuales son:

- Planificación de la Calidad de Software (CS).
- Control de la CS.
- Aseguramiento de la CS.
- Mejora de la CS.

1.3.1- Planificación de la Calidad del Software.

Según la Norma ISO 9000:2000, la planificación de la calidad es la parte de la gestión de la calidad enfocada al establecimiento de los objetivos de la calidad y a la especificación de los procesos operativos necesarios y de los recursos relacionados para cumplir los objetivos de calidad. (ISO 9000:2000).

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

La Planificación de la Calidad del Software es la parte de la Gestión de la Calidad encargada de realizar el proceso administrativo, de desarrollar y mantener una relación entre los objetivos y recursos de la organización; y las oportunidades cambiantes del mercado. (ISO/IEC 90003:2004).

Los aspectos a considerar en la Planificación de la CS son: Modelos/Estándares de CS a utilizar, Costos de la CS, Recursos humanos y materiales necesarios, etc. Los factores que determinan el Modelo o Estándar de CS a elegir son:

- La complejidad del proceso de diseño.
- La madurez del diseño.
- La complejidad del proceso de producción.
- Las características del producto o servicio.
- La seguridad del producto o servicio.
- Económico.

Según la Norma ISO/IEC 90003:2004 se puede decir que:

“La planificación de la calidad facilita el modo de adaptar la planificación del sistema de gestión de la calidad a un proyecto específico, producto o contrato. La planificación de la calidad puede incluir referencias genéricas y/o proyecto / producto / contrato específico de procedimientos, como apropiados. La planificación de la calidad debería ser revisada de nuevo junto con el progreso del diseño y desarrollo, y los elementos, en cada fase, deberían ser completamente definidos al comienzo de dicha fase”.

La planificación de la calidad del software a nivel de proyectos debería considerar lo siguiente:

- Inclusión de los planes de desarrollo.
- Los requisitos de calidad relacionados con los productos y/o procesos.
- Los sistemas de gestión de la calidad adaptando y/o identificando los procesos e instrucciones específicos, apropiados para el ámbito del manual de calidad y algunas exclusiones expuestas.
- Los procesos de proyectos-específicos e instrucciones, tales como, especificación de pruebas del software, detallando los planes, diseños, casos de pruebas y procesos para la unidad, integración, sistemas y pruebas de aceptación.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Los métodos, modelos, herramientas, convenios de lenguajes de programación, bibliotecas, marcos de trabajo y otros componentes reutilizables para ser usados en los proyectos.
- Los criterios para el comienzo y el final de cada fase o etapa del proyecto;
- Los tipos de análisis y otras verificaciones y actividades de validación para ser llevadas a cabo.
- Los procesos de gestión de la configuración para ser llevados a cabo.
- Las actividades de seguimiento y las medidas para ser llevadas a cabo.
- Las personas responsables de aprobar los procesos de salida para su uso posterior.
- La formación necesaria para el uso de herramientas y técnicas, y la organización de la formación previa a la habilidad necesaria.
- Los registros para ser mantenidos.
- La gestión de cambios, como por ejemplo, para recursos.
- Escalas de tiempo y cambios de contrato.

La planificación de la calidad, sin embargo, abreviada es particularmente útil para limitar los objetivos de calidad para el software siendo designados para un propósito limitado. Según (Humphrey ,1989) un plan de calidad puede tener la siguiente estructura:

- Introducción al Producto: Una descripción del producto, su objetivo en el mercado y expectativas de calidad del producto.
- Planes del producto: Fechas críticas respecto de la liberación del producto y responsabilidades del producto respecto de su distribución y servicio.
- Descripciones del proceso: Procesos de desarrollo y servicios que serían usados en el desarrollo y en la administración.
- Objetivos de Calidad: Objetivos y planes de calidad del producto, los cuales incluyen la identificación de los atributos de calidad del producto.
- Manejo del riesgo: principales riesgos que pueden afectar la calidad del producto.

Esta información puede ser presentada en diferentes documentos. El plan de calidad define los atributos de calidad más importantes del producto a ser desarrollado y define el proceso de evaluación de la calidad. En la Planificación de la Calidad del Software se debe determinar:

- Rol de la Planificación.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Requerimientos de la CS.
- Preparación de un Plan de CS.
- Implementación de un Plan de CS.
- Preparar un Manual de Calidad.

1.3.2- Control de la Calidad del Software.

Según la Norma ISO 9000:2000, el control de la calidad es la parte de la gestión de la calidad orientada al cumplimiento de los requisitos de la calidad.

El Control de la Calidad del Software son las técnicas y actividades de carácter operativo, utilizadas para satisfacer los requisitos relativos a la calidad, centradas en 2 objetivos fundamentales: (Pressman, 2002).

- Mantener bajo control un proceso.
- Eliminar las causas de los defectos en las diferentes fases del ciclo de vida.

Está formado por actividades que permiten evaluar la calidad de los productos de software desarrollados. El aspecto a considerar en el Control de la Calidad del Software es la "Prueba del Software".

Las pruebas de software presentan una interesante anomalía para el Ingeniero del Software. Durante las fases anteriores de definición y de desarrollo, el Ingeniero intenta construir el software partiendo de un concepto abstracto y llegando a una implementación tangible. Luego, llegan las pruebas. El Ingeniero crea una serie de casos de prueba que intentan "demoler" el software construido. De hecho, las pruebas son uno de los pasos de la Ingeniería del Software que se puede ver como destructivo en lugar de constructivo.

1.3.3- Aseguramiento de la Calidad del Software.

Según la Norma ISO 9000:2000, el aseguramiento de la calidad es la parte de la gestión de la calidad orientada a proporcionar confianza en que se cumplirán los requisitos de calidad.

El Aseguramiento de Calidad del Software es el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza que el software satisfará los requisitos dados de calidad.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Este aseguramiento se diseña para cada aplicación antes de comenzar a desarrollarla y no después.

El aseguramiento de la calidad del software engloba:

- Un enfoque de gestión de calidad.
- Métodos y herramientas de Ingeniería del Software.
- Revisiones técnicas formales aplicables en el proceso de software.
- Una estrategia de prueba multiescala.
- El control de la documentación del software y de los cambios realizados.
- Procedimientos para ajustarse a los estándares de desarrollo del software.
- Mecanismos de medición y de generación de informes.

Las revisiones del software son un "filtro" para el proceso de Ingeniería del Software. Esto es, las revisiones se aplican a varios momentos del desarrollo del software y sirven para detectar errores y defectos que pueden ser eliminados. Las revisiones del software sirven para "purificar" las actividades de la Ingeniería del Software que suceden como resultado del análisis, diseño y codificación.

La revisión técnica formal (RTF), a veces llamada inspección, es el filtro más efectivo desde el punto de vista del aseguramiento de la calidad y es un medio efectivo para mejorar la calidad del software.

El defecto se define como una anomalía del producto. Dentro del contexto del proceso del software, los términos defecto y fallo son sinónimos. Ambos implican un problema de calidad que es descubierto después de entregar el software a los usuarios finales.

El objetivo principal de las RTF es encontrar errores durante el proceso, de forma que se conviertan en defectos después de la entrega del software. El beneficio de estas RTF es el descubrimiento de errores al principio para que no se propaguen al paso siguiente del proceso de software.

Las actividades de diseño introducen entre el 50 y 65% de todos los errores durante el proceso de software. Sin embargo, se ha demostrado que las RTF son efectivas en un 75% a la hora de detectar errores. Con la detección y la eliminación de un gran porcentaje de errores, el proceso de revisión reduce substancialmente el coste de los pasos siguientes en las fases de desarrollo y mantenimiento.

Los objetivos de la RTF son:

- Descubrir errores en la función, la lógica o la implementación de cualquier representación del software.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Verificar que el software bajo revisión alcanza sus requisitos.
- Garantizar que el software ha sido representado de acuerdo con ciertos estándares predefinidos.
- Conseguir un software desarrollado en forma uniforme.
- Hacer que los proyectos sean más manejables.

La RTF sirve para promover la seguridad y la continuidad, ya que varias personas se familiarizarán con partes del software que, de otro modo, no hubieran visto nunca. Es una clase de revisión que incluye recorridos, inspecciones, revisiones cíclicas y otro pequeño grupo de evaluaciones técnicas del software. Cada RTF se lleva a cabo mediante una reunión y solo tendrá éxito si es bien planificada, controlada y atendida.

El aseguramiento de calidad se refiere a validar los procesos usados para crear los productos. Es una herramienta especialmente útil para administradores y patrocinadores, ya que permite discutir los procesos usados para crear los productos para determinar si son razonables. Este aseguramiento tiene asociado 2 constitutivos diferentes: los Ingenieros de Software que realizan el trabajo técnico y un grupo de Aseguramiento de la Calidad de Software (ACS) que tiene la responsabilidad de la planificación de aseguramiento de la calidad, supervisión, mantenimiento de registros, análisis e informes.

Las Actividades del ACS son:

- Establecimiento de un plan de ACS para un proyecto.
- Participación en el desarrollo de la descripción del proceso de software del proyecto.
- Revisión de las actividades de Ingeniería del Software para verificar su ajuste al proceso de software definido.
- Auditoría de los productos de software designados para verificar el ajuste con los definidos como parte del proceso del software.
- Asegurar que las desviaciones del trabajo y los productos del software se documentan y se manejan de acuerdo con un procedimiento establecido.
- Registrar lo que no se ajuste a los requisitos e informar a sus superiores.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Además de estas actividades, el grupo de ACS coordina el control y la gestión de cambios y; ayuda a recopilar y analizar las métricas del software.

Las métricas son escalas de unidades sobre las cuales puede medirse un atributo cuantificable.

Cuando se habla de software se refiere a la disciplina de recopilar y analizar datos basados en mediciones reales de software, así como a las escalas de medición. Los atributos son características observables del producto o del proceso de software, que proporciona alguna información útil sobre el estado del producto o sobre el progreso del proyecto. El término producto se utiliza para referirse a las especificaciones, a los diseños y a los listados del código. Los valores de las métricas no se obtienen sólo por mediciones. Algunos valores de métricas se derivan de los requisitos del cliente o de los usuarios y, por lo tanto, actúan como restricciones dentro del proyecto.

Los principios básicos de la medición son:

- Los objetivos de la medición deberían establecerse antes de empezar la recopilación de datos.
- Todas las técnicas sobre métricas deberían definirse sin ambigüedades.
- Las métricas deberían obtenerse basándose en una teoría válida para el dominio de aplicación.
- Hay que hacer las métricas a medida para acomodar mejor los productos y procesos específicos.
- Siempre que sea posible, la recopilación de datos y el análisis debería automatizarse.
- Se deberían aplicar técnicas estadísticas válidas para establecer las relaciones entre los atributos internos del producto y las características externas de la calidad.

Las Razones que justifican la Medición del Software son:

- Para indicar la calidad del producto.
- Para evaluar la productividad de la gente que desarrolla el producto.
- Para evaluar los beneficios (en términos de productividad y calidad) derivados del uso de nuevos métodos y herramientas de Ingeniería de Software.
- Para establecer una línea base de estimación.
- Para ayudar a justificar el uso de nuevas herramientas o formación adicional.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Las actividades del proceso de medición son:

- **Formulación:** Obtención de medidas y métricas apropiadas para la presentación del software.
- **Colección:** Mecanismo empleado para acumular datos necesarios para obtener las métricas formuladas.
- **Análisis:** Cálculo de las métricas y aplicación de herramientas matemáticas.
- **Interpretación:** La evaluación de los resultados de las métricas en un esfuerzo por conseguir una visión interna de la calidad de la presentación.
- **Retroalimentación:** Recomendaciones obtenidas de la interpretación de métricas y técnicas transmitidas al equipo de desarrollo de software.

Las medidas de Calidad del Software deben comenzar desde la especificación y terminar con la implementación, implantación y mantenimiento o post- implantación. Debe aplicarse a lo largo de todo el proceso de Ingeniería de Software. Básicamente, la medición es una fase normal de cualquier actividad industrial Sin mediciones es imposible perseguir objetivos comerciales normales de una manera racional.

Existen métricas a nivel de Proyecto, Proceso y Producto respectivamente. Las métricas del Proyecto se consolidan para crear métricas de proceso que sean públicas para toda la organización del software. El uso de métricas para el Proyecto tiene 2 aspectos fundamentales:

- Minimizar la planificación del desarrollo haciendo los ajustes necesarios que eviten retrasos y reducir problemas/riesgos potenciales.
- Evaluar la calidad de los productos en el momento actual y cuando sea necesario, modificando el enfoque técnico que mejore la calidad.

1.3.4- Mejora de La Calidad del Software.

Según la Norma ISO 9000:2000, la mejora de la calidad es la parte de la gestión de la calidad orientada a aumentar la capacidad de cumplir con los requisitos de la calidad. Los requisitos pueden estar relacionados con cualquier aspecto tal como la eficacia, la eficiencia o la trazabilidad.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

La Mejora de la Calidad del Software es la parte de la Gestión de la Calidad que contribuye, por medio de las mediciones, a los análisis de los datos y auditorías, a efectuar mejoras en la calidad del software.

Una Auditoría de Calidad tiene como objetivo mostrar la situación real para aportar confianza y destacar las áreas que pueden afectar adversamente esa confianza. Otro objetivo consiste en suministrar una evaluación objetiva de los productos y procesos para corroborar la conformidad con los estándares, las guías, las especificaciones y los procedimientos.

Las razones para realizar una auditoría son:

- Establecer el estado del proyecto.
- Verificar la capacidad de realizar o continuar un trabajo específico.
- Verificar qué elementos aplicables del programa o Plan de Aseguramiento de la Calidad han sido desarrollados y documentados.
- Verificar la adherencia de esos elementos con el programa o Plan de Aseguramiento de la Calidad.

El propósito y la actividad de la auditoría es recoger, examinar y analizar la información necesaria para tomar las decisiones de aprobación. La auditoría es realizada de acuerdo con los planes y procedimientos documentados. El plan de auditoría establece un procedimiento para dirigir la auditoría y para las acciones de seguimiento sobre las recomendaciones de la auditoría. Al realizar la auditoría, el personal de la misma evalúa los elementos del software y los procesos para contrastarlos con los objetivos y criterios de las auditorías, tales como contratos, requerimientos, planes, especificaciones o procedimientos, guías y estándares.

Los resultados de la auditoría son documentados y remitidos al director de la organización auditada, a la entidad auditora, y cualquier organización externa identificada en el plan de auditoría. El informe incluye la lista de elementos no conformes u otros aspectos para las posteriores revisiones y acciones. Cuando se realiza el plan de auditoría, las recomendaciones son informadas e incluidas en los resultados de la auditoría.

La auditoría puede traer como consecuencia la Certificación. Dicho Proceso de Certificación comienza con la emisión de una Solicitud de Certificación y culmina con la Concesión del Certificado. Un sistema

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

de certificación de calidad permite una valoración independiente que debe demostrar que la organización es capaz de desarrollar productos y servicios de calidad.

En un software se tienen las siguientes visiones de la calidad:

- Necesaria o Requerida: La que quiere el cliente.
- Programada o Especificada: La que se ha especificado explícitamente y se intenta conseguir.
- Realizada: La que se ha conseguido.

El objetivo es conseguir que las tres visiones coincidan. A la intersección entre la calidad Requerida y la calidad Realizada se le llama Calidad Percibida, y es la única que el cliente valora. Toda aquella calidad que se realiza pero no se necesita es un gasto inútil de tiempo y dinero.

La calidad, como sistema de gestión de una organización, necesita definir estos procesos y medirlos, para poder gestionarlos, es decir, para tener la capacidad de proponer mejoras y reconocerlas.

Para implementar un programa de mejoras es necesario definir procesos, decidir qué se quiere mejorar, definir qué medidas serán necesarias recoger, cómo y dónde tomarlas, gestionarlas mediante herramientas, utilizarlas para la toma de decisiones y reconocer las mejoras. Cuando el proceso a mejorar es el de desarrollo del software, es importante definir qué objetivos se quieren alcanzar, para reducir el número de medidas y, en consecuencia, el coste de recopilarlas y el impacto sobre la actividad de producción de software.

La calidad es necesario que forme parte de los productos o servicios que se comercializan para los clientes. El cliente es el mejor auditor de la calidad, él exige el nivel que está dispuesto a pagar por ella, pero no más. Por tanto, debe de cuantificar cuál es el nivel de calidad que nos exige para poder planificar la calidad de los productos que se generen a lo largo de la producción del producto o servicio final. Al analizar las necesidades de los clientes, debe tener en cuenta la previsible evolución de sus necesidades y tendencias en cuanto a características. Debe tener en cuenta la evolución tecnológica del entorno de producción de los productos para suministrarlos con el nivel tecnológico adecuado. No debe olvidar el nivel de calidad de los competidores, debiendo elaborar productos cuyas características y funcionalidades sean competitivas con las de los competidores.

La Calidad de Software es resultado del movimiento global dentro del proceso de mejoramiento continuo de los modelos y/o estándares de producción en todos los sectores industriales, en particular, cuando éste se concentra en la producción de sistemas de información y software especializado.

1.4- Gestión de Pruebas.

Un correcto ciclo de vida de desarrollo software, debería incorporar para cada una de sus actividades, las tareas de verificación y validación, que permitan certificar la corrección de los entregables que se genera.

Existen muchos tipos de pruebas de software (unitarias, funcionales, de carga, stress, rendimiento, seguridad, de regresión, etc.). La mayoría de ellas, requiere para su desenvolvimiento, del uso de herramientas que permitan su automatización. Algunas deberán formar parte de la plataforma de pruebas, por su uso continuado, mientras que otras sólo se necesitaran durante momentos puntuales, variando esta casuística de organización a organización.

Este servicio, tiene como principal objetivo diseñar la metodología de gestión de pruebas, como parte del ciclo de vida del proceso de desarrollo software de las organizaciones.

Al igual que con los servicios orientados a la optimización de los otros procesos (requisitos, gestión de cambios...), se puede establecer objetivos adicionales, tales como:

- Estudio y comparación de herramientas de gestión de pruebas, en base a funcionalidades y costes, para sugerir y decidir cuáles se adaptan mejor a sus necesidades.
- En base a las herramientas elegidas, automatizar e integrar los procesos de pruebas.
- Validar y verificar la metodología y procesos diseñados, sobre un proyecto piloto.
- Proceder al despliegue de los procesos de pruebas, que incluya la formación necesaria a los equipos de analistas, calidad, desarrolladores, etc., así como la instalación y configuración de las herramientas seleccionadas.

1.4.1- Pruebas.

Las pruebas de software presentan una interesante anomalía para el Ingeniero del Software. Durante las fases anteriores de definición y de desarrollo, el Ingeniero intenta construir el software partiendo de un concepto abstracto y llegando a una implementación tangible. Luego, llegan las pruebas. El Ingeniero crea una serie de casos de prueba que intentan “demoler” el software construido. De hecho, las pruebas son uno de los pasos de la Ingeniería del Software que se puede ver como destructivo en lugar de constructivo.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

La prueba es el proceso de ejecutar un programa con intención de encontrar defectos. Es un proceso destructivo que determina el diseño de los casos de prueba y la asignación de responsabilidades. (Pressman, 2002).

La prueba exitosa es aquella que descubre defectos. El "caso de prueba bueno" es aquel que tiene alta probabilidad de detectar un defecto aún no descubierto. El "caso de prueba exitoso" es aquel que detecta un defecto aún no descubierto.

La prueba no es:

- Demostración que no hay errores.
- Demostración que el software desempeña correctamente sus funciones.
- Establecimiento de confianza que un programa hace lo que debe hacer.

La prueba demuestra hasta qué punto las funciones del software parecen funcionar de acuerdo con las especificaciones y parecen alcanzarse los requisitos de rendimiento. Además, los datos que se van recogiendo a medida que se lleva a cabo la prueba proporcionan una buena indicación de la confiabilidad del software e indican la calidad del software como un todo. Pero, la prueba no puede asegurar la ausencia de defectos; sólo puede demostrar que existen defectos en el software.

La prueba del software es un concepto más amplio que, a menudo, es conocido como verificación y validación (V&V). La verificación se refiere al conjunto de actividades que aseguran que el software implementa correctamente una función específica. Su pregunta asociada es: ¿está construyendo el producto correctamente?

La validación se refiere a un conjunto diferente de actividades que aseguran que el software construido se ajusta a los requisitos del cliente. Su pregunta asociada es: ¿está construyendo el producto correcto? La validación del software se consigue mediante una serie de pruebas de caja negra que demuestran la conformidad respecto de los requisitos del cliente. Un plan de prueba traza las clases de pruebas que se han de llevar a cabo, y un procedimiento de prueba define los casos de prueba específicos en un intento por descubrir errores de acuerdo con los requisitos.

Una estrategia de prueba de software integra las técnicas de diseño de casos de prueba en una serie de pasos bien planificados que dan como resultado una correcta construcción del software. La estrategia proporciona un mapa que describe los pasos que hay que llevar a cabo como parte de la prueba, cuándo se deben planificar y realizar esos pasos, y cuánto esfuerzo, tiempo y recursos se van

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

a requerir. Cualquier estrategia de prueba debe incorporar la planificación de la prueba, el diseño de casos de prueba, la ejecución de las pruebas; y la agrupación y evaluación de los datos resultantes.

Las características generales de las estrategias de prueba de software son las siguientes:

- La prueba comienza en el nivel módulo y trabaja "hacia fuera", hacia la integración de todo el sistema basado en computadora.
- Diferentes técnicas de prueba son apropiadas en diferentes momentos.
- La prueba la realiza el que desarrolla el software y un grupo de prueba independiente.
- La prueba y la depuración son actividades, pero la depuración se puede incluir en cualquier estrategia de prueba.

Para implementar con éxito una estrategia de prueba de software, se debe:

- Especificar los requisitos del producto de manera cuantificable antes que comiencen las Pruebas.
- Especificar los objetivos de prueba de manera explícita.
- Desarrollar un plan de prueba que haga hincapié en la prueba de ciclo rápido.
- Construir un software robusto diseñado para probarse a sí mismo.
- Usar revisiones de técnicas formales efectivas como filtro antes de la prueba.
- Realizar revisiones de técnicas formales para evaluar la estructura de la prueba y los propios casos de prueba.
- Desarrollar un enfoque de mejora continua al proceso de prueba.

Cualquier producto de ingeniería puede probarse de una de estas 2 formas:

- Prueba de caja negra.
- Prueba de caja blanca.

La prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. Los casos de prueba pretenden demostrar que las funciones del software son operativas, que la

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene. Este tipo de prueba examina algunos aspectos del modelo fundamental del sistema sin tener mucho en cuenta la estructura lógica interna del software. Los métodos de prueba de la caja negra se centran en los requisitos funcionales de software. La prueba de la caja negra permite al Ingeniero del Software obtener conjuntos de condicionales de entrada que ejerciten completamente todos los requisitos funcionales de un programa. La prueba de la caja negra no es una alternativa a las técnicas de prueba de la caja blanca. La prueba de la caja negra intenta encontrar errores de las siguientes categorías:

- Funciones incorrecta o ausente.
- Errores de interfaz.
- Errores en estructuras de datos o en acceso a bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.

La prueba de caja blanca del software se basa en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones y/o bucles. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado o mencionado. Para este tipo de prueba, se deben definir todos los caminos lógicos y desarrollar casos de prueba que ejerciten la lógica del programa. La prueba de caja blanca no se debe desechar como impracticable. Se pueden elegir y ejercitar una serie de caminos lógicos importantes. Se pueden comprobar las estructuras de datos para verificar su validez. Se pueden combinar los atributos de la prueba de caja blanca y de caja negra, para llegar a un método que valide la interfaz del software y asegure el correcto funcionamiento interno del software. La prueba de caja blanca, denominada prueba de caja de cristal, es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba.

1.4.2- Objetivos de las pruebas.

Entre los objetivos más específicos de las pruebas se encuentran:

- La prueba es el proceso de ejecución de un programa con la intención de descubrir un error.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Un buen caso de prueba es aquel donde se tiene una alta probabilidad de encontrar un error, no descubierto.
- Una prueba tiene éxito si descubre un error no detectado hasta entonces.
- Una prueba contiene una colección de casos de prueba, esto último, especifica la forma de probar el sistema, incluyendo la entrada a probar, resultado esperados y las condiciones bajo las cuales ha de probarse; se tiene que un caso de prueba es exitoso porque posee una alta probabilidad de encontrar un defecto desconocido. (PRESSMAN, 2002).

1.4.3- Características de las pruebas.

El rol responsable de ejecutar las pruebas debe tener conocimiento de que una buena prueba cuenta con las siguientes características (PRESSMAN, 2002):

- Una buena prueba tiene alta probabilidad de encontrar un error. Para lograr esta característica en una prueba es necesario que el responsable sea capaz de entender el software e intentar crearse una imagen de cómo podría fallar.
- Una buena prueba no debe ser redundante: No se deben realizar pruebas que tengan el mismo propósito, pues las pruebas conllevan gran esfuerzo y sin asegurar que los diseños de las mismas no garantizan los objetivos trazados inicialmente se está mal gastando recursos y personal, es decir, todas las pruebas diseñadas deben tener propósitos diferentes, por muy sutiles que estos sean.
- Una buena prueba debería ser la mejor de su cosecha: Se puede llegar a la situación, de que en el diseño de las pruebas se tengan pruebas con propósitos similares y por limitaciones en tiempo y recursos se tenga que reducir el número de pruebas a ejecutar, en este punto es donde se debe escoger la prueba que mayor probabilidad posea de encontrar un error.
- Una buena prueba no debe ser ni muy sencilla ni muy compleja: En determinados casos de prueba se pueden combinar una serie de pruebas de distintos tipos, esto podría enmascarar errores secundarios, por eso, se debe realizar cada prueba por separado.

Es válido aclarar que las pruebas no se ejecutan con la finalidad de demostrar si el sistema no tiene errores, o demostrar si el software realiza correctamente todas sus funcionalidades, o para asegurar si

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

el programa realiza sus funciones, se ejecutan con la finalidad de encontrar errores no detectados en el software.

Incluso, se da una medida de hasta qué punto el software funciona de acuerdo a las especificaciones requeridas. Los datos archivados de los resultados de las pruebas indican en una buena parte, la calidad del software como un todo, pero bajo ninguna circunstancia, las pruebas pueden asegurar la ausencia de defectos; sólo demuestran si el software contiene errores.

1.4.4- Principios de las pruebas.

El diseño de las pruebas de software requiere de amplios conocimientos en las técnicas de diseño de casos de prueba, estas técnicas ayudan a construir casos de pruebas poseedores de altas probabilidades de encontrar un defecto, pero antes de desarrollar esta actividad, el diseñador de pruebas debe conocer los principios básicos de las pruebas: (PRESSMAN, 2002)

- A todas las pruebas se les debería poder hacer un seguimiento hasta los requisitos del cliente: Como anteriormente se ha explicado el objetivo fundamental de las pruebas es encontrar errores, también se puede decir que los errores más graves son aquellos que impiden que el sistema cumpla con los requisitos establecidos.
- Las pruebas deberían planificarse antes de ejecutarlas: La planificación se puede comenzar desde etapas tempranas en el proceso de desarrollo, tan pronto estén los requisitos se puede comenzar a planificar pruebas; los casos de prueba se definen cuando esté consolidado el diseño del sistema, en conclusión se pueden planificar y diseñar todas las pruebas antes de llegar a la implementación.
- El principio de Pareto es aplicable a la prueba del software: En pocas palabras, el principio de Pareto es que del 80% de todos los errores encontrados durante las pruebas se les puede hacer un seguimiento hasta un 20% de todos los módulos del programa, el problemas se encuentra en como aislar los módulos sospechosos y probarlos concienzudamente.
- Las pruebas deberían comenzar por lo pequeño y avanzar hacia lo grande: Las primeras pruebas que se deben ejecutar son las que se centran generalmente en los módulos individuales del programa, así sucesivamente se va subiendo para encontrar errores en grupos integrados y finalmente en el sistema completo.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- No son posibles las pruebas exhaustivas: Aunque sea un programa pequeño el número de caminos que pueden ser ejecutados bajo ciertas condiciones puede ser grande, es por ello, es imposible ejecutar todas las combinaciones de caminos durante las pruebas. Sin embargo, si es posible, cubrir adecuadamente la lógica del programa y asegurarse de que se han aplicado todas las condiciones a nivel de componente.
- Las pruebas deben realizarse por un equipo independiente: El desarrollador que es el responsable de crear su propio código, no es el más recomendable para que pruebe lo que hizo, porque para sus efectos siempre estaría correcto, es por ello, que es aconsejable que las pruebas la realice un equipo independiente, pues el ingeniero del software se limitaría a probar que su programa está libre de errores y no, que realmente contiene los mismo, que al final es el objetivo principal de realizar las pruebas del software.

Las pruebas a ejecutar para probar si el producto responde a los requisitos del cliente con calidad, deben estar bien planificadas y deben tener una estrategia integradora de las técnicas de diseño de casos de prueba, en un conjunto de pasos bien planificados para dar como resultado una buena construcción del software.

Las pruebas de aceptación, forman parte del proceso de validación del software. Muchos tienden a confundir los términos de validación y verificación, por falta de conocimiento acerca del tema. La validación del software incluye un conjunto de actividades que tienen como propósito brindarle los elementos necesarios al cliente para decidir la evaluación final del producto, es decir, si el software será rechazado o aceptado.

Para concluir esta primera parte, se desarrollan dos temas paralelamente, el tema de proceso de validación del software y pruebas de software, al interceptar estos dos momentos surge lo llamado como pruebas de aceptación del cliente.

1.5- Pruebas de Aceptación.

La prueba de aceptación es una de las pruebas que se encuentra en el ciclo de desarrollo del software y su objetivo es el de verificar formalmente con el cliente que el sistema satisface todas sus necesidades.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

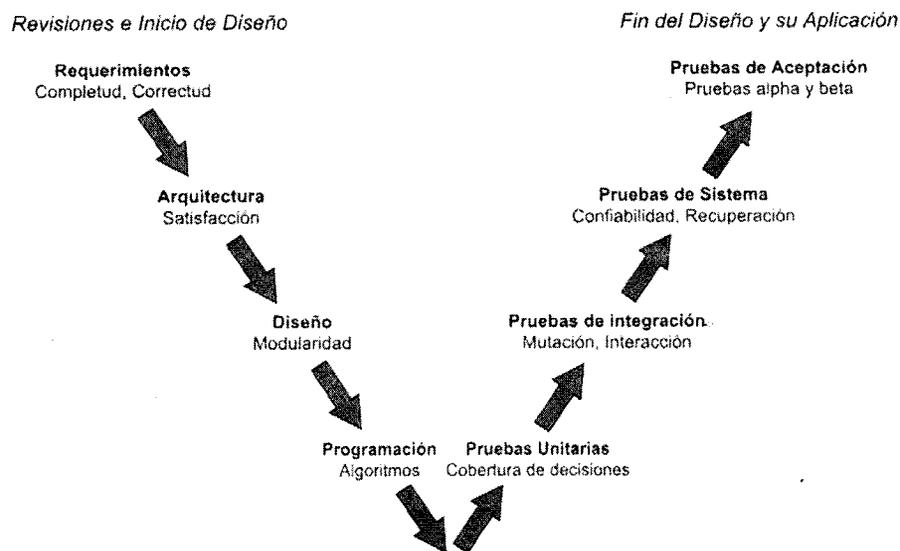


Figura 1- Niveles de prueba con las fases de desarrollo de software. (Pressman, 2002).

Estas pruebas tienen como objetivo validar el producto, verificar si este atiende a los requisitos especificados. Las pruebas de aceptación involucran pruebas funcionales y de sistema. En el caso de las pruebas funcionales se realizan pruebas con el método de caja negra utilizando técnicas de participación equivalentes, análisis del valor límite, de comparación, etc.

Las pruebas de aceptación también involucran pruebas de desempeño, de stress, de seguridad, pruebas para los datos persistentes haciendo en este caso pruebas de frecuencia de uso, restricciones de acceso, restricciones de integridad, requisitos de guarda y retención de datos, etc.

Las pruebas de aceptación son las pruebas ejercitadas por el cliente en su entorno y se realizan con el objetivo de validar si el producto software cumple con las funcionalidades requeridas. Estas pruebas tienen vital importancia pues los resultados proyectados deciden la aceptación o el rechazo del producto. Las pruebas de aceptación surgen por la necesidad de productos software con calidad, capaces de cumplir con las expectativas y los requisitos de los clientes finales.

A continuación se muestra cómo es el proceso de pruebas de aceptación:

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

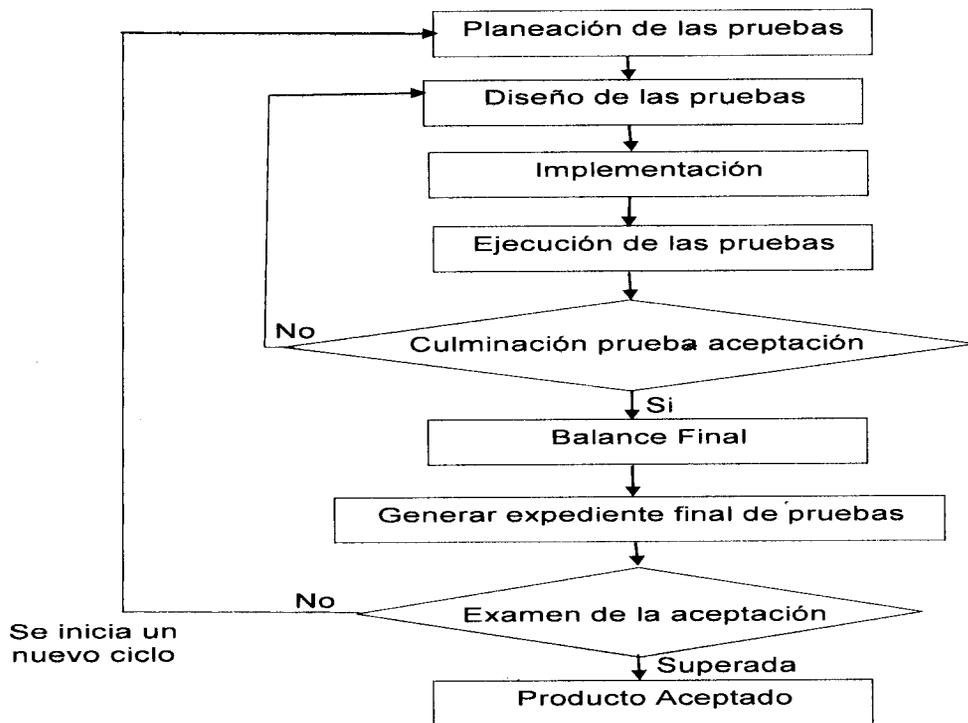


Figura 2- Flujo de trabajo de trabajo de Pruebas de Aceptación. (Yamilis Fernández Pérez, Rosalía Lucia Cué Delgado, Claribel Lucy Cruz Aguila, Michael González Jorrín, Dialaxis Acosta Molina, Violen Hernández Aguilar, 2007).

1.- Planeamiento de las pruebas.

El primer paso es definir y documentar los objetivos de las pruebas, el alcance de las mismas, los elementos a probar, los recursos necesarios, el cronograma y los riesgos en un plan de pruebas.

Este plan de pruebas puede sufrir cambios durante todo el ciclo y es el encargado de guiar las pruebas.

2.- Diseñar pruebas:

A partir de las especificaciones en el plan de pruebas y de las estrategias a seguir se determina: Diseñar los niveles de pruebas, tipo de prueba, procedimiento y caso de pruebas.

Seleccionar herramientas automáticas de pruebas.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Ordenar la ejecución de los casos de pruebas.

En los casos de pruebas se documenta la secuencia de pasos para ejecutar el caso de prueba, los requisitos de datos para realizar las pruebas, los resultados esperados y los métodos para registrar los resultados.

3.- Implementación de las pruebas.

En esta actividad se realiza la instalación del hardware, software y herramientas necesarias para realizar las pruebas. Además se selecciona el equipo de probadores y se capacita al personal.

4.- Ejecución de las pruebas.

Se realizan las pruebas planificadas, tal y como se documenta en los casos de pruebas. Se registran los resultados y se comprueban. Se documenta cualquier desviación de los resultados esperados. Se realizan diariamente reuniones para revisar las no conformidades (NC) registradas y se determina las prioridades de las no conformidades.

5.- Balance final

En esta actividad se realiza el análisis final del ciclo de pruebas de aceptación, elaboración de los informes finales e identificación de artefactos reutilizables.

6.- Examen de la aceptación

Durante el examen de la aceptación, el equipo de desarrollo, clientes y los terceros se reúnen para discutir si se va a aceptar o no el producto. Si el examen es satisfactorio, se acepta el producto si no es satisfactorio se revisan y corrigen los problemas y se pasa a una segunda iteración de este proceso de pruebas.

1.5.1- Definición de las pruebas de aceptación.

Las pruebas de aceptación tienen como fin la validación de ratificar si el sistema cumple con los requisitos requeridos por el cliente y permite al usuario determinar la aceptación del sistema. Por este motivo, las pruebas son realizadas por el cliente y el desarrollador participa de forma pasiva facilitando

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

la comunicación del cliente con el sistema. Son básicamente pruebas funcionales, sobre el sistema completo, y buscan una cobertura de la especificación de requisitos y del manual del usuario.

El sistema es probado con los datos proporcionados por el cliente más que con los datos de prueba simulados. Debido a la diferencia existente entre los datos reales y los datos utilizados en las pruebas, las pruebas de aceptación revelan errores y omisiones en la definición de requerimientos del sistema. También revelan donde los problemas de requerimientos, en los recursos del sistema, no cumplen las necesidades del usuario o donde el desempeño del sistema es inaceptable. (MYERS, 2004).

Durante la ejecución de las pruebas de aceptación el cliente debe plantearle al equipo de desarrollo todos los defectos, fallas y errores que se detecten, antes de validar finalmente el sistema.

Es importante señalar, los criterios de evaluación de las pruebas de aceptación son establecidos por el cliente, precisamente porque es quien realmente tiene los conocimientos de cómo usará el sistema en realidad, puede darse el caso donde los desarrolladores nunca imaginen como será usado el sistema por el usuario final, por eso es una tarea difícil saber lo que se debe probar realmente.

Las pruebas de aceptación tienen objetivos diferentes a las pruebas ejecutadas por los desarrolladores, por sus características diferentes.

1.5.2- Características de las pruebas de aceptación.

Las pruebas de aceptación tienen las siguientes características (JURISTO, 2006):

- Participación del usuario.
- Está enfocada hacia la prueba de los requisitos de usuario especificados.
- Está considerada como la fase final del proceso para crear una confianza en que el producto es el apropiado para su uso en explotación.

Es recomendable que estas pruebas se ejecuten en el entorno en el cuál se va a explotar el sistema.

Para la generación de los casos de prueba de las pruebas de aceptación funcionales se utilizan las pruebas de caja negra. (MYERS, 2004).

La experiencia muestra, después del más cuidadoso proceso de pruebas por parte del desarrollador, quedan una serie de errores que sólo aparecen cuando el cliente se pone a usarlo y para nada es

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

conveniente subestimar el conocimiento del usuario, pues esto puede traer grandes y serios inconvenientes.

1.6- Flujo de trabajo para pruebas definidos por Pressman.

“El flujo de trabajo es la automatización de un proceso de trabajo, en todo o en parte, durante la que documentos, información o tareas son pasadas de un participante a otro por acción, según una serie de reglas de procedimiento”.(La Junta de Castilla y León, 2001).

La prueba es un aspecto muy importante para la calidad del software. Los costos asociados a los errores originan la definición y aplicación de un proceso de pruebas detallistas y bien planificadas. Las pruebas admiten validar y verificar el software, concibiendo como validación el proceso que determina si el producto satisface los requisitos expuestos por el cliente, y verificación el proceso que determina si los artefactos generados en una fase satisfacen las condiciones de dicha fase. Las pruebas tienen como objetivo asegurar la calidad del software desarrollado, además, de que implica la verificación de la interacción de componentes, integración adecuada de los componentes, verificación de que todos los requisitos se han implementado correctamente, identificación y aseguramiento de que los defectos encontrados se han corregido antes de entregar el software al cliente, diseño de pruebas que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo. La prueba no es tan sencilla como parece, no es simplemente una etapa del proyecto en la cual se asegura la calidad, sino que debe ocurrir durante todo el ciclo de vida del producto, debido que desde la fase de inicio se puede hacer una parte de la planificación de las pruebas probando la funcionalidad de los primeros prototipos. La prueba es un proceso que se centra en la lógica interna del software y las funciones externas y tiene la única intención de descubrir un error, por eso se dice que una prueba tiene éxito si descubre un error no detectado hasta entonces. La prueba no puede asegurar la ausencia de defectos; sólo puede demostrar que existen defectos en el software.

Según Pressman las pruebas tienen como intención demoler el software construido, pues es uno de los pasos de la ingeniería del software que se puede ver como destructivo en lugar de constructivo. También plantea algunos objetivos reflejados en el epígrafe 1.4.2 del capítulo 1.

Si una prueba se realiza de acuerdo con los objetivos anteriores tiene una alta probabilidad de encontrar errores. Las pruebas se basan en principios definidos por Pressman que guían al ingeniero para llevar a cabo este proceso, dichos principios están planteados en el epígrafe 1.4.4 del capítulo 1.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Para la realización de una prueba exitosa y más fácil existe un grupo de características:

Operatividad: "Cuanto mejor funcione más eficiente se puede probar."

- El sistema tiene pocos errores.
- Ningún error bloquea la ejecución de las pruebas.
- El producto evoluciona en fases funcionales.

Observabilidad: "Lo que ves es lo que pruebas."

- Todos los factores que afectan los resultados están visibles.
- Un resultado incorrecto se observa fácilmente.
- Se informa automáticamente de los errores internos.

Control habilidad: "Cuanto mejor se pueda controlar el software, más se puede automatizar y optimizar."

- Todos los resultados se pueden generar a través de una combinación de entrada.
- Todo el código es ejecutable a través de una combinación de entrada.

Capacidad de descomposición: "Controlando el ámbito de las pruebas, se podrá aislar mas rápidamente los problemas y llevar a cabo mejores pruebas de regresión."

- El sistema software está construido con módulos independientes.
- Los módulos del software se pueden probar independientemente.

Simplicidad: "Cuanto menos haya que probar, mas rápidamente se podrá probarlo."

- Simplicidad funcional.
- Simplicidad estructural.

Estabilidad: "Cuanto menos cambios, menos interrupciones a las pruebas."

- Los cambios del software son infrecuentes.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Los cambios del software están controlados.

Facilidad de comprensión: "Cuanta más información se tenga, más inteligentes serán las pruebas."

- El diseño se ha entendido perfectamente.
- Se han comunicado los cambios del diseño.

Éstas son características que propician un mejor desarrollo de las pruebas. Unas buenas pruebas según Pressman son la que tienen los atributos planteados en el epígrafe 1.4 del capítulo 1.

Como se ha visto estos son principios, características y aspectos importantes dentro del proceso de pruebas definido por Pressman que conllevan a un desarrollo de pruebas exitoso y con una mayor calidad.

1.7- Conclusiones Parciales.

En este capítulo se ha recopilado toda la información existente a través de los métodos científicos que se han aplicado y lograr que todos puedan adquirir más conocimiento en varios temas importantes como son las pruebas y la calidad de software. Con toda la teoría analizada, se pretende realizar una buena investigación dentro de la Universidad; de cómo se está realizando este proceso de pruebas. También se arribó a conclusiones acerca de la calidad del software, siendo esta un factor importante en la aceptación del software. Dentro de la calidad es muy importante el proceso de pruebas, ya que es la forma idónea de medir la calidad de un sistema, entre las pruebas se encuentra la prueba de aceptación que es la idea esencial de la investigación.

CAPÍTULO 2: ANÁLISIS DE OTRAS SOLUCIONES.

2.1- Introducción.

La calidad tiene un costo asociado y las pruebas como proceso para controlar la calidad consumen tiempo, personal y otros recursos, como en todo proyecto de cualquier índole, siempre se debe tratar que las fallas sean mínimas y al menor costo posible. La fase de pruebas añade valor al producto que se maneja: todos los programas tienen errores y la fase de pruebas los descubre; ese es el valor que añade. El objetivo principal de la fase de pruebas es encontrar la mayor cantidad de errores y defectos.

En este capítulo se pretende analizar los procesos que forman parte en el desarrollo de las pruebas de aceptación en la Universidad de las Ciencias Informáticas (UCI) con la intención de realizar una serie de observaciones que servirán de ayuda y guía a la vez para desarrollar y llevar a cabo una propuesta que perfeccionará el desarrollo de las pruebas.

Se ha desarrollado un análisis de todos los trabajos y tesis que han sido efectuadas en la UCI, para encontrar deficiencias y errores que se puedan corregir, los cuales se van a sustentar de entrevistas que han sido desarrolladas a líderes de proyectos de calidad y de esta forma llegar a un posible resultado. Dichos resultados fusionados con un conjunto de procesos y metodologías y la comparación de las actividades precisadas por Pressman y las que se desarrollan en la UCI, conducen a la creación de una guía para la realización de las pruebas de aceptación en el proceso de desarrollo del software.

2.2- Proceso de calidad en la UCI.

En la UCI el proceso de calidad está estructurada de la siguiente manera:

Existen diez facultades, cada una de ellas posee un proyecto de calidad que se encarga de realizar las pruebas, a su vez todos estos grupos de calidad de las facultades responden y se encuentran interrelacionados con la dirección de calidad de software de la IP, ya que les asigna tareas y los provee de cursos y adiestramientos en el tema de la calidad. La dirección de la IP está vinculada con el centro de calidad de software CALISOFT, pues prácticamente todos los productos que se hacen son enviados allá para que se le realicen las pruebas de aceptación por parte del cliente.

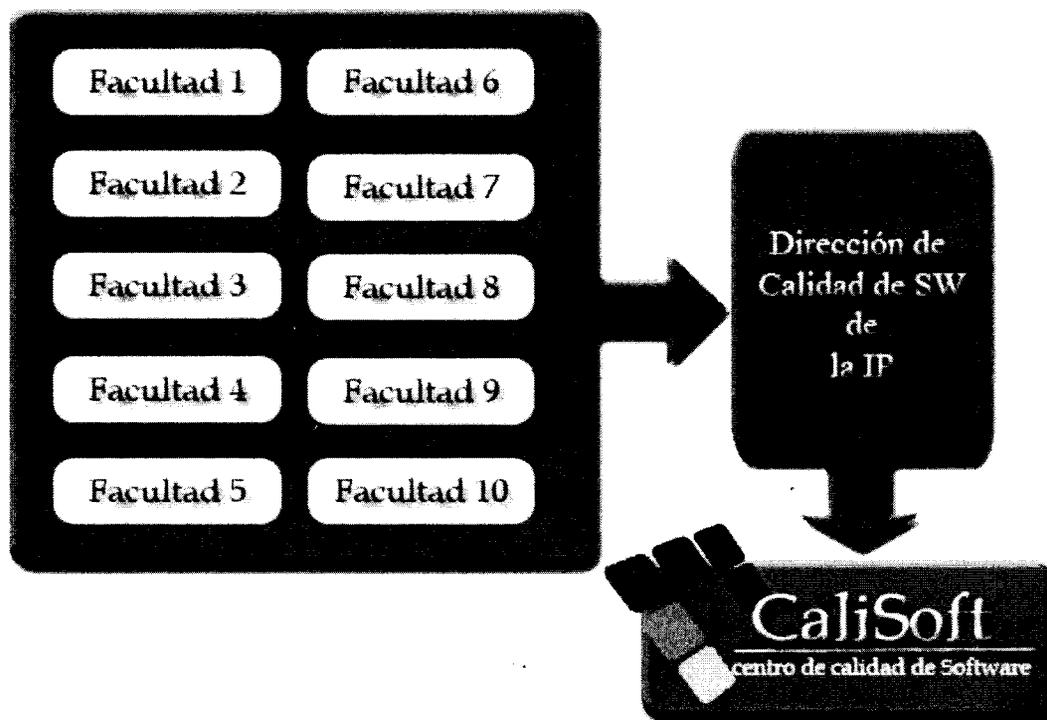


Figura 3- Estructura de la calidad en la UCI.

CALISOFT: Centro de Calidad del Software.

Este centro se encuentra relacionado con diferentes entidades, por ejemplo:

- Centro integrador de Universidades.
- Empresas.
- Centros de Investigación para la mejora de procesos.
- Adscrito al Ministerio de Informática y las Comunicaciones (MIC).

Se dedica a la realización de las pruebas de aceptación con el cliente, pruebas piloto con el usuario final y de la liberación de los entregables y producto final, contribuye al desarrollo de la industria de software en el país incrementando su competitividad a través de la difusión, la mejora continua y el conocimiento de la tecnología de la información. Posee un filtro de calidad para los productos y servicios de software que se adquieran para ser implantados en las organizaciones del país y su principal objetivo es evaluar la aplicación de estándares y modelos de calidad.

CAPÍTULO 2: ANÁLISIS DE OTRAS SOLUCIONES

CALISOFT tiene expectativas para mejorar la calidad de software:

- Desarrollar un programa efectivo de mejora de procesos en el desarrollo de software.
- Extender la gestión de la calidad hacia todos los procesos medulares de la Universidad y a los de apoyo que sustentan el esquema de producción.
- Creación de la Red Temática de Calidad y Mejora de Procesos.
- Difundir el estado del arte y de la práctica, tanto de la ingeniería de software como de la calidad en la misma.
- Certificación Internacional de especialistas cubanos.

2.3- Estado de las pruebas en el mundo.

Por los años de 1970, una regla muy conocida era, que en un proyecto de programación típico, más del 50 por ciento del costo total, era utilizado en las pruebas. (NOTENBOOM, 2003)

Aproximadamente 25 años más tarde la historia se repite. En la actualidad hay nuevos sistemas de desarrollo, lenguajes y desarrolladores que están acostumbrados a desarrollar más rápidamente. Pero en la actualidad las pruebas están adquiriendo un rol importante en los proyecto de desarrollo de software. (JURISTO, 2006)

No es para nada cierto que las pruebas se han convertido en una ciencia exacta, todavía falta mucho camino por recorrer en ese sentido. De hecho se conoce menos de las pruebas de software que de cualquier otro elemento del desarrollo de un software. Las pruebas dejaron de tener importancia en algún momento del proceso general de construcción de software, lo cual se convirtió en un grave error y se pagaron graves consecuencias por ello, sin embargo están nuevamente jugando un papel importante.

Hoy por hoy, las pruebas de software atraviesan por una situación difícil porque no han tomado un estado final, se puede decir que son fáciles pero también difíciles. Las pruebas de software son más difíciles por la diversidad de lenguajes de programación, sistemas operativos y las plataformas de hardware existentes. Las pruebas de software son más fáciles, en algunos casos, porque los sistemas operativos y software, son mucho más sofisticados y proveen intrínsecamente buenas rutinas de pruebas, que pueden ser incorporadas en las aplicaciones, sin la necesidad que el programador las desarrolle desde cero.

CAPÍTULO 2: ANÁLISIS DE OTRAS SOLUCIONES

Las pruebas realizadas al software son un reto interesante en la vida del desarrollador, por la sencilla razón, que a lo largo de todo el desarrollo del software en cada una de las fases del ciclo de vida, se construye el software, se crea; sin embargo en la fase de pruebas específicamente, se trata de probar que ese programa o parte de programa tiene algún defecto, esta tarea se puede ver desde una perspectiva diferente a lo acostumbrado en la construcción de un software; en vez de construir, se trata de alguna manera, destruir lo realizado, con el fin de encontrar donde está el error del sistema.

Como conclusión, se puede decir, la prueba es el conjunto de procesos o actividades llevados a cabo con el fin, de encontrar defectos en un programa (PRESSMAN, 2002). Pruebas, también es una actividad en la cual un sistema o componentes se ejecutan en circunstancias previamente especificadas, los resultados se observan y registran, y se realiza la evaluación de algún aspecto. Una prueba con éxito es aquella donde la ejecución encontró un defecto, tal como se explica en su objetivo principal.

2.4- El proceso de pruebas en Cuba.

En la actualidad en todos los países se ha visto la necesidad de la existencia de las pruebas de software. Cuba está insertándose en este mercado del software, el cual es muy competitivo, por lo que se requiere una alta calidad del producto final. Las pruebas de aceptación no son un servicio ofrecido de manera integral, en una solicitud que pide el cliente para verificar la eficacia del software, es un reto por el que se está trabajando para que sea mejor cada día, pues han descubierto que la calidad no representa un valor agregado sino una exigencia para permanecer en el mercado internacional. La creación de software solo tiene algunas décadas de desarrollo, por lo que en Cuba no está bien definido este proceso en la actualidad. Al inicio del desarrollo de la industria del software, las pruebas se consideraban sólo una actividad que realizaba el programador para encontrar defectos en sus productos; con el paso de los años se ha determinado la importancia que tienen para garantizar el tiempo, el costo y la calidad del producto, de tal forma que actualmente las pruebas definen un procedimiento mediante el cual se evalúa la funcionalidad del software respecto a los requerimientos establecidos por los usuarios. La realización de pruebas de software es un proceso centrado en el objetivo de encontrar defectos a un software; puede ser por razones de depuración al software o de aceptación del mismo. A pesar que generalmente en el mundo se está de acuerdo en que es mejor prevenir defectos que encontrarlos y corregirlos, la realidad es que actualmente el ser humano es incapaz de producir sistemas libres de defectos. La prueba es un elemento esencial en el desarrollo de software, el cual ayuda a mejorar la calidad de este.

CAPÍTULO 2: ANÁLISIS DE OTRAS SOLUCIONES

Las pruebas no mejoran directamente la calidad del sistema, pero sí lo hacen indirectamente, previendo un panorama claro de las debilidades observadas del sistema y de los riesgos asociados para la organización. Esto permite a las administraciones tomar decisiones respecto a la asignación de recursos para mejorar la calidad del sistema. Para lograr estos objetivos, todo proceso de prueba contiene actividades para planear lo que se necesita, especificando qué debe ser probado. Hay una regla universal, la cual indica que es imposible encontrar todos los defectos y que nunca hay suficiente tiempo, personal o dinero para probar todo. Se deben tomar decisiones sabias de cómo distribuir los recursos disponibles.

2.4.1- Estado de las pruebas de aceptación.

Internacionalmente las pruebas de aceptación se realizan de diversas formas, porque es el cliente el encargado de ejecutarlas. Además, las pruebas de aceptación son planificadas de acuerdo a las características específicas del software a construir.

El Instituto Tecnológico de Informática de Valencia España contiene un grupo llamado SQUaC (Software Quality, Usability and Certification) encargado de realizar tareas de calidad, verificación, testeo y usabilidad de sistemas y software. Este grupo ofrece servicios de testeo de funcionalidad donde se examina si una aplicación software satisface los requisitos funcionales esperados. El objetivo de los servicios de testeo de funcionalidad es revelar problemas y errores en lo concerniente a la funcionalidad del producto software y su conformidad al comportamiento documentado, declarado, expresado o deseado por el usuario. En concreto, investigan si una aplicación es apropiada ofreciendo todas las funcionalidades esperadas por el usuario; ofreciendo la exactitud necesitada; y conformidad con lo escrito en el manual de usuario y posibles estándares. (ITI, 2007).

La metodología seguida se basa fundamentalmente en encontrar errores en la funcionalidad de una aplicación de modo que los clientes no lleguen a encontrar errores. Si existe una especificación o lista de control de requisitos, las pruebas verifican si el software los satisface. En caso contrario, trabajan en conjunto con la empresa para desarrollar una lista de control para el testeo e identificación de las características más importantes a verificar. Los conocimientos adquiridos en el desarrollo de software y técnicas de testeo de software, les permite comprender y examinar una aplicación profundamente. Testean las características y funciones de una aplicación bajo condiciones tanto válidas (pruebas positivas) como inválidas (pruebas negativas). En la actualidad las empresas prestigiosas del software

toman muy en serio la validación de sus productos, pues de ello depende el prestigio de sus productos en el mercado.

La creación de la Universidad de las Ciencias Informáticas (UCI) es una estrategia nueva nacida con el objetivo de informatizar la sociedad cubana, en ella se forman profesionales de la rama informática donde desde sus primeros años de carrera se vinculan a proyectos productivos. Esta posibilidad brinda la opción a los estudiantes poder tener un amplio campo en la investigación. Estos proyectos están distribuidos por todas las facultadas en dependencia del perfil de las mismas.

En sus inicios los estudiantes no tenían experiencia en la producción de software, pero se ha ido adquiriendo conocimiento para crear productos con calidad y la ejecución de pruebas para obtener productos mejores.

Donde único se han desarrollado pruebas de aceptación ha sido en el proyecto Registro y Notaría, el proceso se iniciaba cuando los funcionarios comienzan a probar las funcionalidades, a medida que van detectando errores o incomodidades se van anotando en el documento de no conformidad. Al finalizar los especialistas de CALISOFT se aseguran que las no conformidades fueran descritas correctamente, terminado el período de pruebas se revisa el documento de solicitud de cambio y finalmente se libera el producto.

En la Universidad se lucha por lograr producir software con calidad, uno de los aspectos más perseguidos en el mercado del software.

2.5- Descripción actual del dominio del problema.

En la actualidad todos coinciden en reconocer la necesidad de mejorar la calidad del software para poder ser competitivos y permanecer en el negocio. En la Universidad de las Ciencias Informáticas (UCI) este aspecto se tiene muy en cuenta y para lograrlo a cada software se le realizan pruebas.

Dicha estrategia está conformada por un conjunto de elementos y procedimiento basado en casos de uso, principal artefacto de la metodología RUP para llevar a cabo el proceso de pruebas.

En un trabajo realizado por la Ing. Violena Hernández Aguilar y el Ing. Michael González Jorrín denominado "**Proceso de pruebas de caja negra basado en la descripción de casos de uso**" definen pasos que son indispensables para la realización de las pruebas y que no se deberían omitir en ningún momento: (Violena Hernández Aguilar, Michael González Jorrín, 2007).

- Diseño de casos de uso.

CAPÍTULO 2: ANÁLISIS DE OTRAS SOLUCIONES

- Diseño de casos de prueba.
- Realización de las pruebas.

Estos 3 pasos son muy importantes y se está acuerdo con ellos que toda prueba debe tener presente estos aspectos, pero hay otros también que no se pueden obviar y principalmente hay uno que es fundamental y es el plan de pruebas que es el que estructura como se va a desarrollar el proceso de pruebas y mediante el cual se van a guiar para llevar a cabo estos 3 pasos que propone el trabajo.

La descripción de todos los flujos de trabajo incluidos en el proceso de pruebas está referenciada en el epígrafe 1.4 del capítulo 1, donde señala todos los procesos que intervienen así como los documentos que se generan y herramientas que se necesitan para la ejecución de las pruebas.

La investigación se basa principalmente, para darle solución al problema, en conclusiones arribadas en el "Proceso de pruebas de caja negra basado en la descripción de casos de uso", que plantea:

- Un buen diseño de un caso de prueba de caja negra depende de la información que brinde el caso de uso.
- Apenas se cuente con la descripción de un caso de uso se puede comenzar a diseñar los casos de pruebas que lo validan.
- La realización de las pruebas basadas en casos de usos es un proceso que consta de tres pasos fundamentales: diseño, implementación del método y análisis de los resultados y puede tener varias iteraciones.

Es precisamente lo relacionado con la información que brinde el caso de uso donde gira la investigación y por lo que se está de acuerdo con sus conclusiones, pues todo proceso de prueba que se lleve a cabo debe guiarse esencialmente en esta información.

En la UCI esta tarea es muy importante y se le hace a todo tipo de software que se desarrolle, a continuación les comentare como es el flujo de trabajo.

Primeramente cuando es terminado el software se envía para el lugar donde se le realizarán las pruebas, al llegar también trae incluido los casos de uso y los requisitos del producto para saber que debe cumplir, se diseñan los casos de pruebas a partir de los casos de uso o de los requisitos porque no siempre envían los casos de uso, que es lo principal en este proceso, posteriormente se ejecutan la pruebas, se llena el documento de no conformidad y le entregan los resultados al los desarrolladores para que corrijan los errores.

CAPÍTULO 2: ANÁLISIS DE OTRAS SOLUCIONES

Este flujo de trabajo en la UCI se le hace a todo tipo de software, sin tener en cuenta que características posea el software. Estas pruebas son realizadas por grupos de personas que conforman el proyecto de calidad.

Las pruebas de aceptación no se realizan en la UCI, solo se llevan a cabo en Calisoft. Este centro ha desarrollado un proceso de pruebas que se ha utilizado en la aceptación de varios productos de software de la UCI, con resultados satisfactorios. Ha sido reconocido y bien recibido por parte de los clientes probadores y los equipos de desarrollo que han participado en este proceso.

Para no depender de Calisoft y poder contar con una guía para realizarlas en la escuela, es que ha surgido esta investigación, por todo esto, el principal objetivo es seguir validando y perfeccionando este proceso de pruebas de aceptación y lograr la automatización del mismo para adquirir una mayor satisfacción por parte del cliente realizando una guía metodológica basándose en la información que brindan los casos de uso.

2.6- Situación Problemática.

En la Universidad de las Ciencias Informáticas la calidad se tiene muy en cuenta y para lograr un producto sin errores se realiza el proceso de pruebas que es tan importante en el desarrollo del software. Las pruebas son realizadas por grupos de calidad que son los encargados de encontrar todos los errores que contiene el software.

Para realizar las pruebas fundamentalmente los jefes de proyectos de calidad se basaban en aspectos como requerimientos y casos de uso que son muy importantes en esta tarea.

Después de llevar a cabo un análisis de cómo se estaban desarrollando las pruebas en la UCI, no se obtuvieron los resultados esperados ya que existe una gran necesidad de obtener productos con calidad y en la Universidad no se realizan este tipo de pruebas de Aceptación al cliente, solo se realiza una revisión final, porque no se utilizan correctamente los Casos de Uso, debido a la inexistencia de una guía que permita formalizar una manera adecuada de desarrollarlas.

La universidad se dio la tarea de realizar un estudio profundo para lograr que el proyecto Calidad tuviera una guía para una mejor realización de las Pruebas de Aceptación de un producto con la utilización de los Casos de Uso.

2.7- Flujo de trabajo en la UCI para el desarrollo de la pruebas.

El proceso de pruebas en la UCI se desarrolla en los laboratorios de calidad, y comienzan cuando a este se le asigna el proyecto. Las facultades le comunican al proyecto de calidad la propuesta para realizar las pruebas a un determinado producto, luego el equipo de desarrollo y de calidad se reúnen y planifican los cronogramas de entrega y liberación de cada artefacto, esto será firmado por ambas partes y se elabora el Plan de pruebas, el cual es el encargado de guiar las pruebas, en el queda recogido todo lo referente a las pruebas tanto por parte de los desarrolladores como por parte de los ingenieros de pruebas, todo lo que se menciona en el proceso que se comienza a describir aquí está incluido en el plan de pruebas. Pues este plan de pruebas es el que recoge todo lo referente a estas pruebas. Luego el equipo de desarrollo entrega al equipo de calidad un expediente, este expediente se crea según la norma cubana NC ISO/IEC 12119, este contiene el producto software a probar, el documento de especificación de casos de uso, el manual de usuario y de instalación, un glosario de términos, y en caso de no estar incluidos en la especificación de casos de uso, un documento que contiene los requerimientos y a que caso de uso corresponde cada uno de ellos. También deberán entregar un documento con los usuarios, permisos, nombres y demás detalles necesarios que se deban conocer para la instalación del software y no estén incluidos en los manuales.

El grupo de desarrollo debe garantizar además, un producto estable en el lugar que ha preparado el líder del proyecto de calidad donde se van a realizar las pruebas, durante los días que se vayan a realizar las pruebas, para montar este bien este lugar el equipo de desarrollo deberá entregar al equipo de calidad con antelación los requerimientos mínimos de hardware y software para que la aplicación funcione. Según las necesidades y características del software que se este probando se necesitara además un documento con algunos juegos de datos que contiene la base de datos, con el objetivo de que el ingeniero de pruebas cuente con todo lo necesario y utilice datos específicos. Antes de comenzar las pruebas se crea un expediente del producto, el cual incluye todo lo entregado por los desarrolladores y se le ira incluyendo todo lo que se genere durante el proceso de pruebas. Luego de aplicadas las pruebas se obtiene como resultado de las mismas el documento de no conformidades, en el mismo están todos los defectos, errores y sugerencias realizadas al equipo de desarrollo, así como también queda documentado en el caso de prueba el resultado de todas las pruebas aplicadas. El equipo de calidad entrega estos resultados al equipo de desarrollo, el cual trabajara en función de estos para perfeccionar al máximo el software, este proceso se repetirá cuantas veces sea necesario, y por esto se establecerá una retroalimentación entre los dos equipos de trabajo en cuanto a los defectos encontrados, cuando se entreguen versiones nuevas que solucionen errores señalados por el

CAPÍTULO 2: ANÁLISIS DE OTRAS SOLUCIONES

equipo de calidad el equipo de desarrollo deberá entregar un documento adjunto a la nueva versión, que especifique exactamente que parte se cambio en cada artefacto, para evitar volver a indicar errores ya señalados, y aun no corregidos, este documento puede ser el mismo documento de no conformidades con modificaciones en sus columnas según se plantea en las "Bases para la comunicación con un grupo de desarrollo" De esta forma el grupo de calidad se retroalimenta de las correcciones realizadas y sirve de guía a los desarrolladoras para el cumplimiento con los cambios necesario que implican las no conformidades. Al terminar el proceso se tendrá el expediente actualizado con todo lo correspondiente al proceso de pruebas, así quedara archivado un historial sobre las pruebas realizadas, este incluirá un informe sobre las pruebas realizadas, este informe recoge todo lo sucedido en las diferentes etapas del proceso de pruebas.

El flujo de trabajo antes mencionado fue elaborado por Yanet Fernández Pons de la Facultad 3, que pertenece al proyecto de calidad, en el trabajo llamado "Descripción del proceso de pruebas".

Está muy completo e integrado a la vez, solo que le faltaron pequeños detallitos y existen cierto desorden a la hora de realizar las tareas, o sea el orden definido no es el adecuado.

2.8- Entrevistas realizadas.

Las entrevistas tienen como principal objetivo escuchar la opinión de expertos en el área para de manera comunicativa relacionarnos con el tema de la investigación.

2.8.1- Análisis de la entrevista a Ramsés Delgado.

Según la entrevista realizada a Ramsés Delgado cuando un software es liberado por el proyecto de Calidad UCI; este pasa a Calisoft donde un grupo de expertos son los encargados de certificar el software como listo para hacerle la entrega al cliente. Un vez que este lo tiene en sus manos él decide si realiza las pruebas de aceptación o no. Mayormente es probada una parte del sistema, es decir un módulo o algunos casos de uso más importantes para el cliente. Al preguntar se informa que los clientes no tenían una guía para realizar estas pruebas, ellos solamente cuentan con un plan de pruebas que se realizó antes con un representante de Calisoft, el Gerente de proyectos en Cuba y el Gerente de proyectos de Venezuela que es el país que en la actualidad Cuba tiene relaciones, donde aquí quedan fijados varios acuerdos y es aquí donde Cuba en varias ocasiones no cumple en la fechas de encuentros con el cliente, porque unos de las deficiencias que se están dando es la mala planificación en las pruebas, porque si el proyecto de Calidad UCI se tarda en liberar el proyecto la

CAPÍTULO 2: ANÁLISIS DE OTRAS SOLUCIONES

entrega a Calisoft se atrasa. Cuando el cliente tiene el software en sus manos, si él decide hacer las pruebas de aceptación, junto a él se encuentran uno o varios expertos en el tema. El cliente es quien escoge en todo tiempo lo que se va ser aprobado, por tanto la UCI no tiene dominio de cuanto quiere probar el cliente es por esto que se pretende mostrarle al cliente en un futuro una guía por la cual se pueda guiar cuando quiera realizar estas pruebas. Si el encuentra una No-Conformidad en lo que está siendo probado en ese momento y es sencilla, uno de los expertos quizás la pueda solucionar en este mismo instante, por ejemplo si es un documento de Manual de Usuario o alguna descripción de los Casos de Uso, etc., es fácil para el experto, pero si es más complicado y no es capaz de darle una solución, se planifica otro encuentro donde ya se le presentaría la solución a la No-Conformidad y si posible solucionar por el grupo de proyecto, porque en ocasiones el cliente pide algo que no queda bien explícito en los requisitos funcionales y después es engorroso de informatizar algo que está dudoso. Según los datos recogidos en la entrevista el cliente es el principal para realizar las pruebas de aceptación.

2.8.2- Análisis de la entrevista a Dialexis Acosta Medina.

En la entrevista que se le hizo a Dialexis Acosta Medina se le preguntaron aspectos interesantes y de gran importancia para la realización de la pruebas de aceptación. En sus respuestas quedaron plasmadas como se desarrollaba el proceso actualmente.

Las pruebas de aceptación se realizan si el cliente lo solicita y en ese caso quien las prepara y las lleva a cabo, es la entidad certificadora a la cual ellos lo soliciten, en estos momentos en Venezuela se la están solicitando a CALISOFT, que es la cara comercial, de la Dirección de Calidad de Software en la UCI. No existe una guía formal para hacerlas, solo se basan en normas y estándares, también se apoyan en el Pressman y artículos de internet.

A estas pruebas se le da suma importancia, pues como se sabe son las últimas que se le realizan al software antes de ser liberado, principalmente por la parte certificadora (CALISOFT) y por la parte del cliente en su garantía del producto.

Cuando es encontrada una no conformidad se procede de manera normal, como está definido en la UCI y como se realiza en todos los proyectos de calidad, se le dan a conocer a los desarrolladores las no conformidades (NC), y realizan gestión de cambio a medida que las van reparando para las iteraciones.

CAPÍTULO 2: ANÁLISIS DE OTRAS SOLUCIONES

Esta prueba genera artefactos como por ejemplo, plan de pruebas, documento de no conformidad, solicitud de cambio, entre otros.

Los roles principales que intervienen son, los funcionarios (Personas de parte de los clientes que se conozcan el negocio o que sean los que van a trabajar con la aplicación). Los especialistas de calidad que presiden las pruebas y el equipo de desarrollo.

2.9-Investigaciones desarrolladas en la UCI acerca de pruebas de aceptación.

Desde hace décadas, se conoce que el mayor esfuerzo dedicado a un software se realiza después de entregado a sus clientes, entre un 60 y 80% (PRESSMAN, 2002). Las pruebas de aceptación se ejecutan luego de culminar la fase de pruebas de sistema, esta ejecución se realiza en la última etapa de pruebas, la cual evaluará el sistema final con miras a su presentación frente al cliente. En esta prueba se verificará si el programa cumple con las especificaciones formales establecidas por el cliente. El objetivo principal de dichas pruebas es la determinación por parte del cliente de la aceptación o rechazo del sistema desarrollado.

En la Universidad de las Ciencias Informáticas (UCI) se han desarrollado varios trabajos relacionados con el tema y que plantean los problemas y riesgos que se corren a la hora de ejecutar las pruebas de aceptación. Estos trabajos han aportado grandes beneficios al proceso de desarrollo del software. Los riesgos son un aspecto que no se puede obviar a la hora de desarrollar cualquier tipo de software, pues en cada etapa existen riesgos diferentes, por lo que hay que tenerlos muy en cuenta.

“Un riesgo es un evento o condición incierto que si ocurre, tendrá un efecto positivo o negativo en los objetivos del proyecto”. (Project Management Institute A Guide to the Project Management Body of Knowledge).

En el trabajo titulado “**Análisis de riesgos involucrados en las pruebas de aceptación con el cliente**”, realizado por: Violena Hernández Aguilar, Michael González Jorrín y Ailyn Febles Estrada se plantean los principales riesgos que se cometen a la hora de realizar las pruebas de aceptación. (Violena Hernández Aguilar, Michael González Jorrín, Ailyn Febles Estrada, 2007)

Riesgos relacionados con la documentación entregada por parte del equipo de desarrollo:

Los Manuales de Usuario e Instalación tienen mala calidad y existe retraso en la entrada de los manuales al proceso de pruebas.

CAPÍTULO 2: ANÁLISIS DE OTRAS SOLUCIONES

Los Requisitos Funcionales y no funcionales: Nunca fueron escritos para guiar el desarrollo y se escriben al final y la comprensión de la necesidad del cliente no se recoge correctamente.

Riesgos relacionados con el cronograma de pruebas:

El calendario de pruebas y los cronogramas de liberación de software son poco realistas y acurren excesivas actualizaciones dentro del proceso de pruebas que no están contempladas en el plan de pruebas.

Riesgos relacionados con los recursos humanos involucrados en las pruebas. (Probadores, Desarrolladores y CALISOFT).

Riesgos relacionados con el personal (Riesgos de los Recursos). No hay suficiente personal disponible para la realización de las pruebas.

Organización del proceso: No se definen los líderes y no se establecen los canales de documentación adecuados.

Dependencia de los recursos de los terceros y de la organización de los proyectos.

Riesgos relacionados con la Documentación generada a partir de la realización de las pruebas.

Los reportes diarios: Se entrega el Informe de No Conformidades. Ocurren repeticiones de una misma no conformidad con otras palabras. Dificultades para acordar un sistema de trabajo que acomode a todos los involucrados.

Estos riesgos se pueden asumir de varias formas, se evitan, se aceptan o se transfieren, si se decide aceptar el riesgo hay tomar alguna decisión para reducir el impacto que cause. Para los realizadores del trabajo su mejor opción es aceptar los riesgos, pues primeramente identificaban los riesgos, los analizan y luego realizan un plan para mitigarlos. Se realizó un resumen de los riesgos más importantes a tener en cuenta en los proyectos y en la realización de las pruebas de aceptación, con el objetivo de tener presente estos riesgos en CALISOFT y en todo lugar donde se realice el proceso de pruebas de aceptación con el cliente.

En otro trabajo realizado en la UCI "**Pruebas de aceptación parciales y finales del cliente: una nueva variante.**", realizado por Geiser Arcio Pérez Riva, Nadiesda Sanz Carmenate, Michael González Jorrín y Ailyn Febles Estrada. (Geiser Arcio Pérez Rivas, Nadiesda Sanz Carmenates, Michael González Jorrín, Ailyn Febles Estrada, 2007).

Los autores establecen una nueva variante para las pruebas de aceptación, donde lo que se quiere es desarrollar las pruebas de aceptación no solo al final del proceso de desarrollo sino de forma parcial.

CAPÍTULO 2: ANÁLISIS DE OTRAS SOLUCIONES

Definieron que al culminar cada módulo se le realizarían las pruebas de aceptación con el objetivo de comprobar si cumple con los requisitos del cliente, al terminar todos los módulos es cuando se desarrollarían las pruebas de aceptación finales, porque las que se hacen antes son parciales.

En cada iteración surgen algunos artefactos: Plan de Pruebas de Aceptación del cliente; el Plan de Pruebas de Aceptación Parcial de cliente – Iteración N; Plan de Pruebas de Aceptación Final del cliente; Casos de Pruebas; Informe de No Conformidades y el Resumen de los Resultados de las Pruebas, así como los resultados finales y cualquier tipo de incidencia ocurrida.

Para las pruebas parciales se llevan a cabo varios pasos: Aprobar el plan de pruebas; Realizar las pruebas de aceptación; Elaborar el informe de no conformidad; Elaborar el resumen de los resultados de las pruebas; Analizar los cambios propuestos.

Para las pruebas finales se llevan a cabo varios pasos: Aprobar el plan de pruebas; realizar las pruebas de aceptación; elaborar el informe de no conformidad y las solicitudes de cambios del cliente; corregir errores en caso de existir y realizar las pruebas de nuevo; elaborar el informe final; liberación del producto.

Para la realización de este trabajo, se hizo un análisis del anterior donde los autores se proponen realizar pruebas parciales y totales y fundamentan que de esta manera el producto sería mejor aceptado por el cliente ya que la posibilidad de que tenga errores se reduciría a lo largo de su desarrollo; pero se decidió no seguir esta propuesta porque no cumple los objetivos propuestos en esta investigación, que no es más que elaborar la guía para la realización de las pruebas de aceptación por parte del cliente al final del desarrollo del software.

2.10- Conclusiones Parciales.

Con la elaboración de este capítulo se alcanza el objetivo principal, que es entender con profundidad los aspectos más importantes para desarrollar la propuesta de solución. Se analiza la situación de las pruebas de software y con mayor especificidad las pruebas de aceptación del cliente en la Universidad de las Ciencias Informáticas (UCI). También, se muestra un conjunto de trabajos realizados en la Universidad que sirven de apoyo para la investigación. Con todo lo expuesto en este capítulo se puede iniciar el paso más importante de la investigación, que no es más que elaborar una guía para desarrollar las pruebas aceptación en los laboratorios de calidad de las facultades de la UCI y en Cuba.

CAPÍTULO 3. PROPUESTA DE SOLUCIÓN PARA EL DESARROLLO DE LAS PRUEBAS DE ACEPTACIÓN

CAPÍTULO 3. PROPUESTA DE SOLUCIÓN PARA EL DESARROLLO DE LAS PRUEBAS DE ACEPTACIÓN.

3.1- Introducción.

En el presente capítulo se describe la definición de la solución a la investigación, se explica como quedó elaborado el flujo de trabajo para el proceso de pruebas de aceptación, los roles que intervienen y los artefactos involucrados. La solución se basa en una guía que puede ser usada en los laboratorios de calidad o en Cuba en caso que hiciera falta. Para elaborar esta guía se consultó todo lo plasmado en los capítulos anteriores, que sirvieron de base para su confección. Las pruebas de aceptación se desarrollarán de una manera estratégica y con la mayor calidad posible, donde los que la realizan poseen habilidades, funciones y actividades a realizar. En el siguiente capítulo se profundiza más en el tema de la solución.

3.2- Pasos para un buen proceso de pruebas.

Las Pruebas de Aceptación se efectúan junto con el cliente para verificar que el Software está listo para ser utilizado por el usuario final, asegurando que el producto desarrollado cumple con sus expectativas.

Para arribar a la solución de la investigación se efectuó un intenso análisis y estudio de todo lo antes mencionado, después de algunas opiniones y fundamentos se concretó una serie de aspectos, por ejemplo: pasos que no se deben olvidar nunca en un proceso de pruebas, artefactos y roles en las pruebas de aceptación y un flujo de trabajo basado en errores de otros existentes.

Antes de llevar a cabo el proceso de pruebas de aceptación es bueno tener en cuenta una serie de pasos que son fundamentales para un buen desempeño de esta actividad:

Identificar el alcance del proceso de pruebas de software que se va a desarrollar.

Este paso es muy importante pues define el tipo de prueba a realizar, la plataforma tecnológica que posee el producto, las distintas etapas del proceso de pruebas, todo lo que se debe entregar con su requerida prioridad así como los criterios y responsabilidades en la verificación de los defectos.

CAPÍTULO 3. PROPUESTA DE SOLUCIÓN PARA EL DESARROLLO DE LAS PRUEBAS DE ACEPTACIÓN

Estructura del equipo de proyecto.

Habitualmente un proyecto donde se realicen pruebas de software existen los siguientes roles:

Líder de proyecto: Es el responsable de dirigir y guiar el desarrollo de las pruebas, analizar los resultados y crear acciones para mejorar.

A la vez el líder de proyecto es el encargado de distribuir los otros roles y asignar las responsabilidades que tiene cada uno.

Funciones:

- Gestiona y asigna recursos humanos y de otro tipo.
- Define las prioridades de las tareas dentro y/o relacionadas con el proyecto.
- Coordina las interacciones con los clientes y los usuarios finales.
- Planifica las iteraciones.
- Planifica y asigna las tareas de la forma más razonable posible.
- Define la organización y estructura del proyecto.
- Establece las líneas de trabajo a seguir para garantizar la calidad e integridad de los artefactos del proyecto.
- Motiva y organiza el equipo de trabajo para lograr un objetivo definido.
- Establece los horarios de trabajo del equipo de desarrollo.
- Planifica y realiza las reuniones de control del equipo de desarrollo en el tiempo establecido.
- Mantiene el control del resultado de estas reuniones.
- Informa sobre el estado actual del proyecto.
- Mantiene el plan del proyecto.
- Suministra al equipo el informe del ciclo de desarrollo.

Habilidades:

- Dominar técnicas de motivación y organización del trabajo en grupo.
- Dominar técnicas de distribución y asignación de recursos humanos y materiales.

CAPÍTULO 3. PROPUESTA DE SOLUCIÓN PARA EL DESARROLLO DE LAS PRUEBAS DE ACEPTACIÓN

- Utilizar métodos de comunicación.
- Tener conocimiento del completo ciclo de vida de un proyecto software según RUP.
- Dominar técnicas de planificación de tiempo, estimación de costos, estimación de proyectos y estimación de riesgos.
- Dominar la organización de equipos de desarrollo de software, estructura de equipos, distribución de tareas y administración del tiempo.
- Dominar los artefactos generados en un proceso de desarrollo de software.
- Dominar los controles de versiones.
- Dominar las herramientas de planificación como el Microsoft Project.
- Tener conocimiento sobre las Metodologías Ágiles de desarrollo de software.
- Dominar las metodologías de investigación.
- Dominar Normas de Calidad y Estándares Internacionales.
- Dominar principios de la Legalidad de la producción de software (propiedad intelectual, licencia de software)

El personal asignado.

La experiencia es fundamental para el proceso de pruebas, por lo que hay que elegir el personal con mayor capacidad y ver si ha participado en otros procesos de pruebas con éxito. También debe contar con una estabilidad laboral para que no exista ninguna interrupción.

Por todo esto es imprescindible que se ofrezca entrenamiento profesional en esta especialidad al personal.

Interacción del grupo con el personal de pruebas.

La visita de un miembro del proyecto a revisar es primordial para de esta forma entender como funciona el negocio y los puntos de vista que existen.

Los miembros del proyecto deben estar en contacto con el grupo de pruebas para acceder a información que pueda ser necesitada en alguna ocasión.

CAPÍTULO 3. PROPUESTA DE SOLUCIÓN PARA EL DESARROLLO DE LAS PRUEBAS DE ACEPTACIÓN

Estos son pasos que hay que tener presentes antes de poner en práctica cualquier proceso de pruebas.

Los roles es algo que no se puede obviar y que hay que tratarlo con mucho cuidado, pues cada uno cumple un función y a la vez realiza actividades que de incumplirlas interrumpiría el proceso de pruebas de aceptación.

3.3- Descripción del Flujo de Trabajo Final que le da solución a la investigación.

A continuación se muestra el flujo de trabajo propuesto, con todas sus especificaciones y artefactos que interactúan en el mismo.

Lo primero que ocurre es asignarle el proyecto al grupo de calidad e informarle cuando comienzan, posteriormente el líder de proyecto organiza un encuentro con los desarrolladores para que expliquen las principales características y mencionen como funciona el negocio. Luego de terminada esta reunión se le asigna a los integrantes de calidad los roles de cada uno y las responsabilidades que ocupan dentro del proceso de pruebas de aceptación.

Una vez definido y organizado todo el trabajo se le pide a los desarrolladores la información necesaria para llevar a cabo el trabajo, donde lo más importante son los casos de uso y sus descripciones, al recibir dicha información se evalúa y se selecciona la más significativa para el proceso y solo la que se necesita para de esta manera disminuir el tiempo de trabajo.

Con toda esta información se procede al planeamiento de las pruebas, donde se dice el tiempo, las fechas de entrega, entre otras antes mencionadas, es precisamente en esta actividad donde se generan los artefactos plan de pruebas y expediente del producto. Al concluir este paso se realiza el diseño de pruebas que de orienta de acuerdo a lo definido en el plan de pruebas y el expediente del producto, aquí surgen los casos de pruebas, artefacto muy importante dentro de este proceso.

Diseñadas las pruebas se ejecutan las mismas por el cliente, donde se definen los criterios de aceptación, que son recogidos precisamente en el artefacto con este nombre, recordar que siempre al cliente lo acompaña un experto. Culminada esta tarea si existe alguna no conformidad que se debe registrar en el documento de no conformidades, se analizan con los desarrolladores, se ponen de acuerdo y se empieza nuevamente el diseño de las pruebas. En caso de no existir ninguna no conformidad se realiza el balance final donde se escribe en el expediente del proyecto el resultado de las pruebas. Al finalizar esto se lleva a cabo el examen de aceptación, donde el cliente da sus puntos de vistas y en caso de cambiar algo lo escribe en el informe de solicitud de cambio y se hacen las

CAPÍTULO 3. PROPUESTA DE SOLUCIÓN PARA EL DESARROLLO DE LAS PRUEBAS DE ACEPTACIÓN

pruebas nuevamente empezando por la planeación, pero si no propone ningún cambio, finalmente se acepta el producto.

3.3.1- Actividades fundamentales.

1- Asignar el proyecto.

Esta actividad se inicia cuando la facultad concluye algún proyecto o se le asigna otro para que le realice las pruebas, de esta forma al proyecto calidad se le entrega el proyecto.

2- Planificar una reunión con los desarrolladores.

Se reúnen los integrantes con lo desarrolladores para que le expliquen como funciona el negocio y el flujo de trabajo del mismo, así como todos los procesos que se encuentran incluidos.

3- Planificar una reunión con los integrantes del proyecto y con el líder de proyecto para asignar roles y responsabilidades.

El líder de proyecto asigna los roles necesarios para el trabajo y las responsabilidades que le corresponden a cada uno.

4- Recoger la información existente.

Los desarrolladores entregan toda la documentación referente al producto, necesaria para el proceso de pruebas, como por ejemplo, casos de uso, manual de instalación, glosario de términos, etc.

5- Analizar la información recogida.

De la información entregada se escoge la más importante para ahorrar tiempo en el proceso y que las pruebas sean más fructíferas.

6- Planear el proceso de pruebas.

Lo principal de esta actividad es definir los objetivos de las pruebas, el alcance, los elementos a probar, el cronograma de trabajo y los riesgos que intervienen. Es aquí donde surge el plan de pruebas que es el encargado de guiar las pruebas y que puede tener cambios en el transcurso del proceso.

CAPÍTULO 3. PROPUESTA DE SOLUCIÓN PARA EL DESARROLLO DE LAS PRUEBAS DE ACEPTACIÓN

7- Diseñar las pruebas.

Se apoya en el plan de pruebas para diseñar los niveles de pruebas, tipos de pruebas y los casos de pruebas, en los casos de prueba se documentan todo lo necesario para realizar las pruebas. Es en este paso donde se tiene muy en cuenta la información brindada por los casos de uso, principal artefacto de entrada a la hora de diseñar los casos de pruebas.

8- Ejecutar las pruebas.

Se realizan las pruebas guiadas por los casos de pruebas, se analizan los resultados obtenidos, y en caso de surgir alguna no conformidad se registra en el documento de no conformidad (NC).

9- Terminar las pruebas.

Son terminadas las tareas dentro del proceso de pruebas.

10- Balance final.

Esta actividad tiene como objetivo analizar el ciclo de las pruebas de aceptación y elaborar los informes finales.

11- Realizar examen de aceptación.

Para esta actividad se reúnen los clientes con el equipo de desarrollo y los expertos para analizar si se va a aceptar el producto, corregir posibles problemas, si es convincente el examen de aceptación se acepta el producto, sino se corrigen los problemas y se realiza una nueva iteración.

12- Aceptar el producto.

Si pasa el examen de aceptación los clientes aceptan el producto y termina el proceso de pruebas de aceptación.

CAPÍTULO 3. PROPUESTA DE SOLUCIÓN PARA EL DESARROLLO DE LAS PRUEBAS DE ACEPTACIÓN

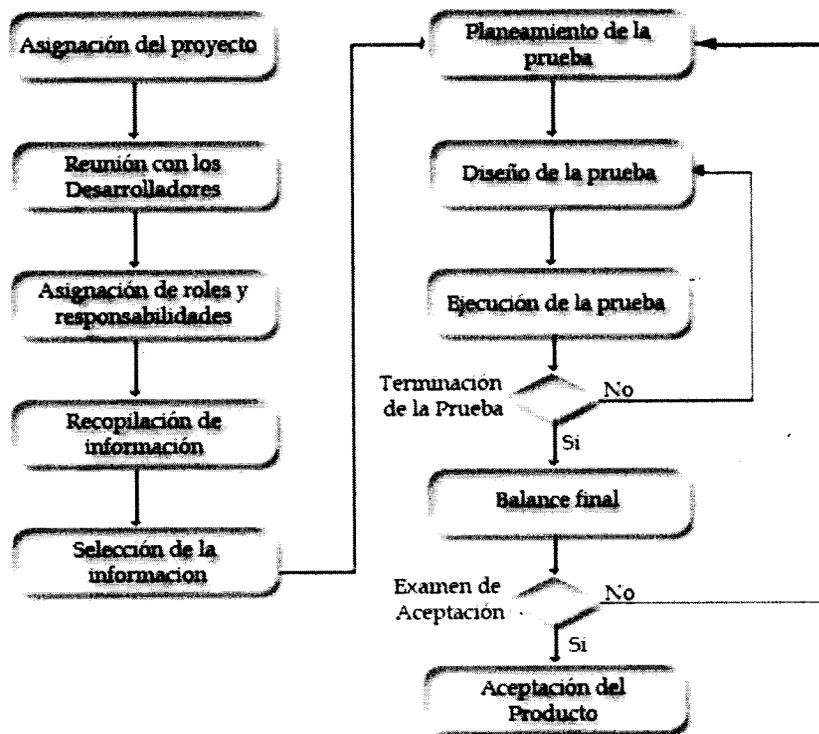


Figura 4- Actividades fundamentales.

3.4- Roles que intervienen en el proceso.

Los roles es algo que no se puede obviar y que hay que tratarlo con mucho cuidado, pues cada uno cumple un función y a la vez realiza actividades que de incumplirlas interrumpiría el proceso de pruebas de aceptación.

El primer rol que debe estar presente es el de **Administrador de Prueba**, pues es el encargado de administrar las pruebas, dirigir y organizar el proceso y del éxito de las mismas. Sus actividades más importantes son las de identificar elementos para facilitar el proceso, evaluar la calidad e informar al líder de proyecto los resultados.

Otro rol es el de **Analista de Prueba** que es responsable de identificar y definir las pruebas requeridas en una iteración, supervisando el progreso de las pruebas, así como el resultado para cada ciclo de prueba. El rol de manera general tiene la responsabilidad de representar las necesidades de los

CAPÍTULO 3. PROPUESTA DE SOLUCIÓN PARA EL DESARROLLO DE LAS PRUEBAS DE ACEPTACIÓN

clientes que no se relacionan directamente con el proyecto. Este rol es una especialización del administrador de prueba. Su tarea principal es la de identificar los objetivos de la prueba.

Hay un miembro del equipo que se encarga de identificar las técnicas apropiadas, herramientas y pautas para llevar a cabo las pruebas y usar los recursos correspondientes para el esfuerzo de las mismas, este es el **Diseñador de Prueba** que se centra en diseñar y describir los casos de prueba.

A la hora de realizar el cliente la prueba ya el rol de **probador** lo toma él mismo, pero existe un miembro del grupo de calidad que lo acompaña siempre por si existe un error o duda, este rol no esta definido, solo que casi siempre es un experto que sabe el funcionamiento del producto.

3.4.1- Descripción de los roles.

Los roles juegan un papel importante a la hora de llevar a cabo el proceso de pruebas de aceptación, pues cada uno tiene funciones, habilidades y actividades a desarrollar que son vitales dentro de su rol.

Administrador de Prueba:

Habilidades:

- Administrar las pruebas, dirigir y organizar el proceso y de el éxito de las mismas.
- Identificar los objetivos de la prueba.
- Utilizar métodos de comunicación.
- Tener conocimiento del completo ciclo de vida de un proyecto software según RUP.
- Dominar técnicas de planificación de tiempo, estimación de costos, estimación de proyectos y estimación de riesgos.
- Dominar el contenido que esta en la creación de la planificación del Plan de pruebas.
- Responsable de guiar las pruebas según lo planificado en el Plan de Pruebas.
- Estar presente ante cualquier cambio que se pueda efectuar.
- Saber realizar anotaciones en el Expediente de proyecto.
- Dominar diseño de casos de pruebas.

Funciones:

- Es responsable del Plan de pruebas.

CAPÍTULO 3. PROPUESTA DE SOLUCIÓN PARA EL DESARROLLO DE LAS PRUEBAS DE ACEPTACIÓN

- Tiene que hacer un resumen de los resultados de las pruebas.
- Debe negociar y acordar las fechas y demás para realizar las pruebas.
- Asegurarse de una correcta planificación y administración de recursos.
- Tiene que verificar el progreso y efectividad de las pruebas.
- Tiene que evaluar los resultados de prueba.

Actividades que debe realizar:

- Trabajar acorde a la misión.
- Identificar los elementos que motiven o faciliten las pruebas.
- Evaluar y auditar la calidad.
- Evaluar y mejorar las pruebas.

Analista de Prueba:

Habilidades:

- Listar ideas de las pruebas.
- Diseñar un modelo de prueba.
- Describir los casos de pruebas.
- Definir los procedimientos de pruebas.
- Coleccionar y administrar datos de pruebas.
- Coleccionar los resultados de las pruebas.
- Evaluar los resultados de cada ciclo de prueba.

Funciones:

- Responsable de identificar y definir las pruebas requeridas.
- Monitorea el progreso de las pruebas y resultado en cada ciclo de pruebas.
- Evalúa la calidad total experimentada como un resultado de las actividades de prueba.

CAPÍTULO 3. PROPUESTA DE SOLUCIÓN PARA EL DESARROLLO DE LAS PRUEBAS DE ACEPTACIÓN

- Responsable de representar apropiadamente las necesidades de los stakeholder que no tienen representación regular y directa en el proyecto.

Actividades que debe realizar son:

- Identificar los objetivos de la prueba.
- Identificar las ideas de la prueba.
- Definir los detalles de las pruebas.
- Definir necesidades de evaluación y trazabilidad.
- Determinar los resultados de las pruebas.
- Verificar los cambios en la construcción.

Diseñador de Pruebas

Habilidades:

- Conocer el proceso de planificación de las pruebas necesarias para cada iteración (pruebas de sistema y pruebas de integración).
- Dominar diseño de casos de pruebas.
- Dominar Herramientas para el control de errores.
- Dominar las Normas y Estándares Internacionales.
- Documentar la prueba de arquitectura.
- Especificar la interfaz de prueba.
- Configurar el ambiente de prueba.
- Definir la suite de pruebas.
- Identificar y describir apropiadas técnicas de prueba.
- Identificar herramientas apropiadas de soporte.
- Definir y mantener la arquitectura de automatización de pruebas.
- Especificar y verificar las condiciones de las configuraciones de prueba.

CAPÍTULO 3. PROPUESTA DE SOLUCIÓN PARA EL DESARROLLO DE LAS PRUEBAS DE ACEPTACIÓN

Funciones:

- Define la estrategia de pruebas y asegurar su correcta implementación.
- Identifica las técnicas apropiadas, herramientas e instrucciones para implementar las pruebas necesarias.
- Diseña las pruebas al software en construcción en sus diferentes etapas del desarrollo.
- Se centra en el diseño de los casos de prueba.

Actividades que debe realizar son:

- Desarrollar la lista de defectos.
- Reportar las no conformidades.
- Definir los enfoques de las pruebas.
- Definir la configuración del ambiente de las pruebas.
- Identificar los mecanismos de pruebas.
- Estructurar la implementación de las pruebas.
- Definir los elementos de las pruebas.
- Desarrollar las pautas de las pruebas.

Probador:

Habilidades:

- Dominar Herramientas para el control de errores.
- Dominar los artefactos generados en un proceso de desarrollo de software.
- Implementar pruebas.
- Implementar la suite de pruebas.
- Ejecutar la suite de pruebas.
- Analizar fallos de prueba.

CAPÍTULO 3. PROPUESTA DE SOLUCIÓN PARA EL DESARROLLO DE LAS PRUEBAS DE ACEPTACIÓN

Funciones:

- Este rol lo toma el mismo cliente, solo que casi siempre es un experto que sabe el funcionamiento del producto y acompaña siempre al cliente.
- Debe conducir las pruebas necesarias.
- Debe reportar las no conformidades en el Informe de No Conformidad en caso de encontrar alguna.
- Es el responsable de realizar las anotaciones en el Informe de Solicitud de Cambio.

Actividades que debe realizar son:

- Test log (salida de ejecución de las pruebas).
- Test Script (Pasos o instrucciones para realizar la prueba).
- Identificar la implementación más apropiada para la prueba.
- Implementar pruebas individuales.
- Identificar las No conformidades.
- Verificar la forma en que se ejecutan las pruebas.
- Analizar y recuperarse de los errores de ejecución.

3.5- Artefactos presentes en la solución.

En este proceso de pruebas de aceptación se definió que se iban a generar los siguientes artefactos:

El principal artefacto que existe es el **plan de pruebas**, que es el encargado de guiar las pruebas y puede tener cambios durante el ciclo, incluye el propósito de las pruebas, qué elemento se va a probar, las herramientas a utilizar y con qué recursos, así como el documento que va hacer entregado. Al tener el resultado de las pruebas se puede comparar lo obtenido con lo esperado. Se reflejan las características de hardware y software que serán empleados para realizar el conjunto de las pruebas al sistema.

Luego de realizado el plan de pruebas se crea el **expediente del producto**, donde se recoge todo lo entregado por los desarrolladores, todo lo que ocurre en el proceso y los resultados de las pruebas, así como la especificación de casos de uso, el manual de usuario y de instalación, un glosario de

CAPÍTULO 3. PROPUESTA DE SOLUCIÓN PARA EL DESARROLLO DE LAS PRUEBAS DE ACEPTACIÓN

términos y los requerimientos y a que caso de uso corresponde cada uno de ellos. También los **Casos de Pruebas** son artefactos que organizan el desarrollo específico de las pruebas de cada caso de uso del módulo, este artefacto define un conjunto de datos de entradas, condiciones de ejecución y resultados esperados de las pruebas, identificados para hacer una evaluación de los aspectos específicos de un elemento objeto de prueba. Cada Caso de Prueba está asociado a un escenario de un Caso de Uso en particular. Los casos de prueba deben ser escritos con el detalle suficiente para que el probador pueda empezar rápidamente a ejecutar pruebas y a encontrar defectos. Además, estos reflejan trazabilidad con los casos de uso, las especificaciones suplementarias de requerimientos y diseño del sistema, garantizando que los procedimientos de pruebas sean compatibles con las necesidades de los usuarios/clientes. En la guía, los Casos de Uso dirigen todo el proceso de desarrollo, es por ello que los Casos de Uso se transforman en un activo que puede directamente conducir el proceso de pruebas. Un Caso de Prueba no es igual a un Caso de Uso. El Caso de Prueba extiende o amplía la información contenida en un Caso de Uso.

Los **criterios de aceptación** determinan la precisión mínima requerida o las características específicas de funcionamiento necesarias para que los resultados obtenidos en las pruebas e inspecciones puedan garantizar la adecuación del producto a sus especificaciones, describe cómo el cliente evaluará los entregables del proyecto y bajo qué condiciones aceptará el producto, incluyendo los casos de pruebas del proyecto a ejecutarse.

Para llevar a cabo las pruebas de aceptación se diseñan **juegos de datos** que se guardan en la plantilla de los juegos de datos, que define los tipos de datos necesarios para el desarrollo de las pruebas, así como ejemplos de ello.

Luego de terminadas las pruebas los errores son guardados en el **Informe de No Conformidades**, encargado de recopilar todas las no conformidades surgidas durante el proceso, y por último el **informe de solicitud de cambio** que es llenado por el cliente, quien es el que realiza estas pruebas y el cual contiene todas las solicitudes de cambios surgidas.

CAPÍTULO 3. PROPUESTA DE SOLUCIÓN PARA EL DESARROLLO DE LAS PRUEBAS DE ACEPTACIÓN

3.5.1- Artefactos más importantes.

Plan de Pruebas.

El propósito del plan de pruebas es explicitar el alcance, enfoque, recursos requeridos, calendario, responsables y manejo de riesgos de un proceso de pruebas. Note que puede haber un plan global que explicita el énfasis a realizar sobre los distintos tipos de pruebas.

La construcción de un buen Plan de Pruebas es la piedra angular y en consecuencia el principal factor crítico de éxito para la puesta en práctica de un proceso de pruebas que permita entregar un software de mejor nivel. No obstante que cada esfuerzo o proceso de pruebas puede ser diferente y específico, la mayor parte de los proyectos informáticos, sean de nuevos desarrollos o de mantenimiento de aplicaciones, tienen un marco común para la realización de las pruebas.

El plan de pruebas es un instrumento muy útil para dar estructura tanto al proceso de pruebas como a la documentación de las mismas. Se distinguen diferentes tipos, que pueden variar desde un plan general, aplicable a todas las pruebas, hasta un plan detallado que se utiliza para un tipo de pruebas específico, como la prueba de aceptación, la prueba de interfaz, etcétera.

Un buen plan de pruebas ayuda a la organización del proyecto completo, a la estimación de los recursos necesarios y a la gestión de la calidad en general.

Un plan de pruebas incluye:

Identificador del plan.

Preferiblemente de alguna forma mnemónica que permita relacionarlo con su alcance, por ej. TP-Global (plan global del proceso de pruebas), TP-Req-Seguridad1 (plan de verificación del requerimiento 1 de seguridad), TP-Contr-X (plan de verificación del contrato asociado al evento de sistema X), TP-Unit-Despachador.Iniciar (plan de prueba unitario para el método iniciar de la clase Despachador). Como todo artefacto del desarrollo, está sujeto a control de configuración, por lo que debe distinguirse adicionalmente la versión y fecha del plan.

Alcance

Indica el tipo de prueba y las propiedades/elementos del software a ser probado.

Elementos a probar

CAPÍTULO 3. PROPUESTA DE SOLUCIÓN PARA EL DESARROLLO DE LAS PRUEBAS DE ACEPTACIÓN

Indica la configuración a probar y las condiciones mínimas que debe cumplir para comenzar a aplicarle el plan. Por un lado, es difícil y riesgoso probar una configuración que aún reporta fallas; por otro lado, si espera a que todos los módulos estén perfectos, puede que detecte fallas graves demasiado tarde.

Estrategia

Describe la técnica, patrón y/o herramientas a utilizarse en el diseño de los casos de prueba. Por ejemplo, en el caso de pruebas unitarias de un procedimiento, esta sección podría indicar: "Se aplicará la estrategia caja-negra de fronteras de la precondición" o "Ejercicio de los caminos ciclomáticos válidos". En lo posible la estrategia debe precisar el número mínimo de casos de prueba a diseñar, por ej. 100% de las fronteras, 60% de los caminos ciclomáticos. La estrategia también explicita el grado de automatización que se exigirá, tanto para la generación de casos de prueba como para su ejecución.

Casos de Pruebas.

El diseño de casos de prueba para el software o para otros productos de ingeniería puede requerir tanto esfuerzo como el propio diseño inicial del producto. Sin embargo, los Ingenieros de Software tratan las pruebas como algo sin importancia, desarrollando casos de prueba que "parezcan adecuados", pero que tienen poca garantía de ser completos. Se deben diseñar pruebas que tengan la mayor probabilidad de encontrar el mayor número de errores con la mínima cantidad de esfuerzo y tiempo posible.

A partir del momento en que el equipo de desarrollo ha elaborado los casos de usos del sistema se pueden elaborar los casos de pruebas que no son más que un conjunto de entradas con datos de prueba, unas condiciones de ejecución, y unos resultados esperados con el propósito de identificar y comunicar las condiciones que se llevarán a cabo en la prueba. Los casos de pruebas son necesarios para verificar la aplicación exitosa y aceptable de los requisitos del producto (que se especifican en los casos de uso).

Los casos de pruebas serían entonces un conjunto de entradas con datos de prueba, unas condiciones de ejecución, y unos resultados esperados cuyo propósito es Identificar y comunicar las condiciones que se llevarán a cabo en la prueba. Los casos de la prueba son necesarios para verificar la aplicación exitosa y aceptable de los requisitos del producto (casos de uso).

El método que se propone para el diseñar los casos de pruebas es una variante del de particiones equivalentes, propuesto por RUP y al que se le han añadido algunas consideraciones, este método

CAPÍTULO 3. PROPUESTA DE SOLUCIÓN PARA EL DESARROLLO DE LAS PRUEBAS DE ACEPTACIÓN

consta de 3 pasos fundamentales y se basa totalmente y usa como artefacto de entrada los casos de usos y opcionalmente el documento de reglas del negocio.

- Para cada caso del uso, generar un sistema completo de escenarios.
- Identificar los Casos de Prueba.
- Identificar Valores de Datos para las Pruebas

Documento de No Conformidad.

El proceso de prueba es un proceso finito por lo que se hace necesario establecer un criterio que indique cuando parar las pruebas tanto en un proceso de liberación como de aceptación, ya sea porque la aplicación efectivamente cumple con todos lo establecido, o porque no lo hace y es necesario entonces volver al flujo anterior (implementación).

Es necesario definir entonces un nuevo concepto, la No Conformidad (NC) que no es más que problemas detectados en un artefacto (casos de usos, sistema, etc.) según:

Error con respecto a lo definido en artefactos anteriores y/o en lo pactado con el cliente.

No concordancia con Normas internacionales que deben ser cumplidas por el artefacto.

Insatisfacción del cliente con el resultado final de un Elemento de Configuración según lo pactado con anterioridad en el proyecto.

(Violena Hernández Aguilar, Michael González Jorrín, 2006).

El impacto de las no conformidades se le asigna un grado de acuerdo a la respuesta que provoque:

3	Alta	Errores que impidan llegar al final del flujo básico. (liberación) Errores en la interpretación del proceso de negocio que impediría el correcto funcionamiento de la aplicación para la consecución del fin. (aceptación del cliente)
---	------	---

CAPÍTULO 3. PROPUESTA DE SOLUCIÓN PARA EL DESARROLLO DE LAS PRUEBAS DE ACEPTACIÓN

2	Media	Errores que impidan llegar al final en un flujo alterno
1	Baja	Errores de validación y ortografía

Figura 5- Grado de las No Conformidades.

Luego de revisar el Documento de No Conformidad el cliente acepta la aplicación según un criterio establecido con el equipo de desarrollo, y mientras este no se cumple se siguen realizando iteraciones de pruebas.

3.6- Diagrama de actividades.

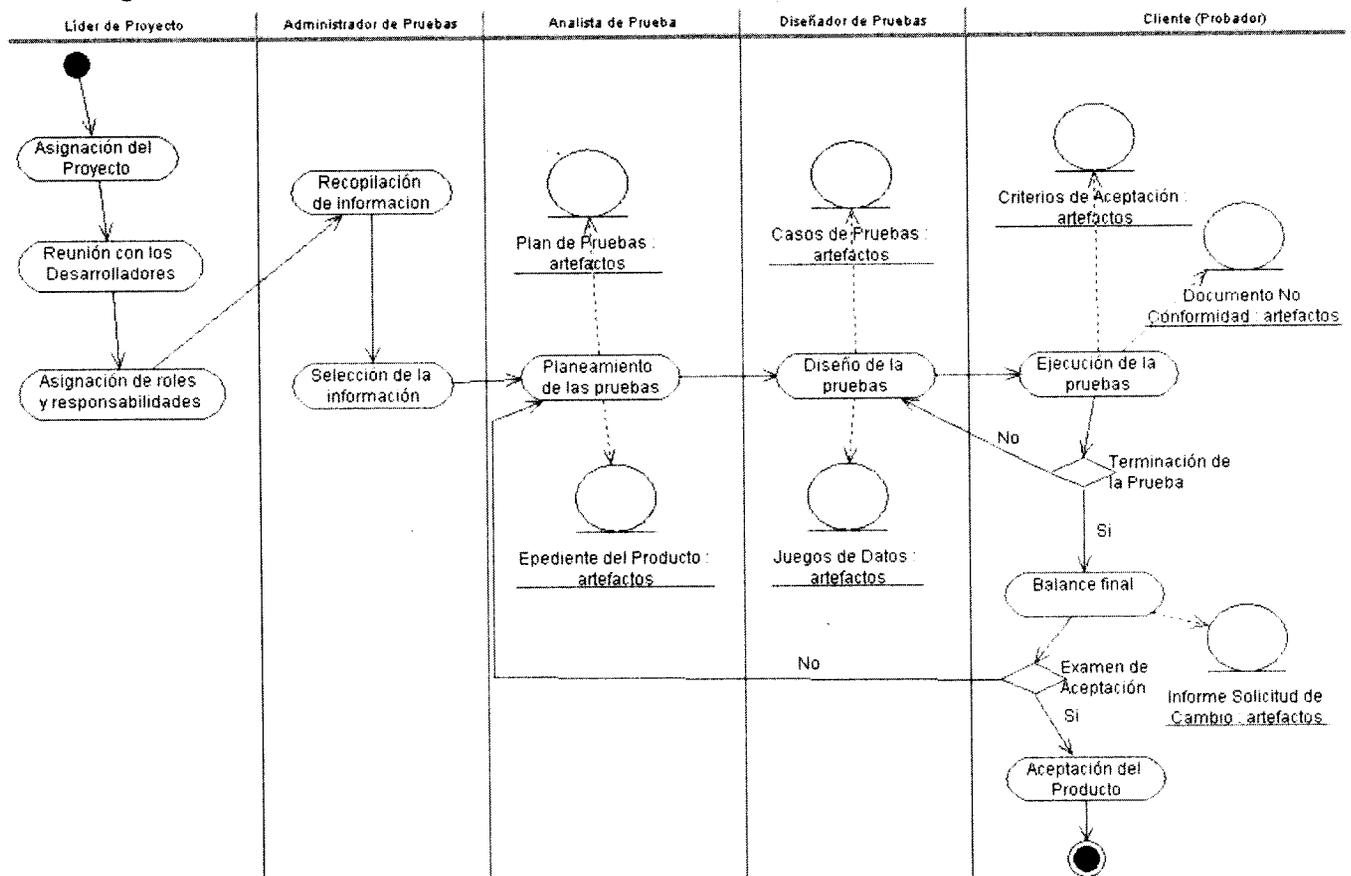


Figura 6- Diagrama de actividades.

CAPÍTULO 3. PROPUESTA DE SOLUCIÓN PARA EL DESARROLLO DE LAS PRUEBAS DE ACEPTACIÓN

3.7- Opiniones de expertos.

La solución propuesta fue enviada a expertos en pruebas de aceptación para de esta forma validarla y observar el criterio acerca de todos los pasos, roles y artefactos que se encuentran presentes. Una de las que aportó su opinión fue Tayche Capote García:

“Considero que en sentido general el proceso propuesto está bien definido. Tiene mucho que ver con lo que se hace normalmente en la UCI en las pruebas, pero más dirigido a la aceptación en el cliente. Los roles definidos tienen una fundamentación clara y cada cual tiene su responsabilidad bien diferenciada, de todas maneras sería bueno evaluar la cantidad de personas, pues es posible que pueda reducirse teniendo en cuenta que no son necesarias tantas personas para guiar la prueba y la parte de diseño de CP puede obviarse, pues pueden reutilizarse los CP que se utilizan en las pruebas de liberación. Es buena una propuesta como esta, pues puede adaptarse para las pruebas de aceptación con los clientes nacionales o de la propia UCI, para lo cual no se hacen las pruebas de aceptación formales que deberían hacerse, pero recomiendo que se alinee mucho más con la forma de hacer las pruebas de Calisoft, pues es una entidad que ya tiene experiencia en este tipo de pruebas.”

3.8- Conclusiones Parciales.

Con la solución planteada en este capítulo se da cumplimiento al objetivo general de la investigación: establecer una guía que permita realizar las Pruebas de Aceptación de un producto utilizando la documentación relacionada con los casos de uso, esta solución se desarrolló de forma íntegra y con la mayor calidad posible. También se le dio solución a los objetivos específicos: elaborar la guía y validarla de forma eficiente. Esta descripción de todos los pasos a seguir a la hora de realizar las pruebas de aceptación, aborda todos los aspectos necesarios para ejecutarlas y reflejó de manera clara aportes a la investigación. Gracias a esto las pruebas de aceptación podrán ser aplicadas en todas las facultades de la Universidad y en Cuba, pues quedó conformada una guía de validación del software para una mejor aceptación del cliente.

CONCLUSIONES

CONCLUSIONES

Con la finalización de la investigación del siguiente trabajo de diploma y el planteamiento de la solución se llegó a conclusiones:

- La documentación analizada sobre la línea de trabajo de pruebas de software facilitó la comprensión de este proceso y sirvió de apoyo para la investigación.
- Las pruebas son un aspecto determinante para la producción, pues de esto depende medir hasta que punto el software cumple con las especificaciones del cliente.
- Se demostró la importancia de contar con una guía para la realización de las pruebas de aceptación, para asegurar que el software no sea rechazado por parte del cliente.
- La solución propuesta se ajusta a los estándares internacionales para el desarrollo de las pruebas de software.

Con la propuesta de solución presentada en la investigación se logró finalizar de manera exitosa el principal objetivo, definir la guía. Para conseguir este objetivo se profundizó en el proceso de validación, se investigó cuales eran los conceptos fundamentales relacionados con las pruebas de aceptación y se analizaron los elementos que se encontraban dentro del proceso de pruebas de aceptación. También se analizó como se trataba el tema en la Universidad y el impacto que provoca.

En conclusión todas las tareas de la investigación fueron cumplidas con los resultados esperados y principalmente se le dio solución al problema principal de trabajo: "Establecer una guía que permita realizar las Pruebas de Aceptación de un producto utilizando la documentación relacionada con los casos de uso", desarrollando las tareas específicas con los resultados esperados.

RECOMENDACIONES

RECOMENDACIONES

Luego de terminada una guía que permite la realización de las pruebas de aceptación con la mayor calidad posible, se hace necesario plantear un grupo de recomendaciones para ir mejorando de forma continua el proceso de pruebas en la UCI.

- Los líderes de proyecto de calidad pueden utilizar el trabajo para enriquecer sus conocimientos sobre pruebas de aceptación.
- Aplicar la guía a un conjunto de software.
- Si son satisfactorios los resultados, proponer la guía a toda la Universidad.
- Proponer el trabajo para que sea utilizado en eventos científicos y como ayuda o base para investigaciones posteriores.
- Dejar el documento como fuente de consulta y bibliografía para el tema de las pruebas de aceptación.

REFERENCIAS BIBLIOGRÁFICAS

REFERENCIAS BIBLIOGRÁFICAS

1. Pressman, R. "Ingeniería del Software: Un Enfoque Práctico". 2002.
2. Juristo, N., Moreno, A. M., and Vegas, S. "Técnicas de evaluación de software", 2006.
3. Myers, G. J. "The art of Software Testing", 2004.
4. Notenboom, B. B. A. E. "Testing Embedded Software", 2003.
5. "Testeo de funcionalidad", 2007.
6. Humphrey, W., "Introducción al Proceso de Software Personal". 2001.
7. ISO 9000:2000, "Sistema de Gestión de la Calidad Principios y Vocabulario", 2000.
8. Antonio, A. d. (1999). Gestión, Control y Garantía de la calidad del Software.
9. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. El Proceso Unificado de Desarrollo de Software. La Habana: Félix Varela, 2004.
10. ISO/IEC 12207:1995, Norma Internacional, Ciclo de Vida del Software.
11. MSc. Yamilis Fernández, I.D.A.M., Ing. Violena Hernández Aguilar, Ing. Michael González Jorrín, Ing. Pérez, Ing. Rosalía Lucia Cué Delgado, Lic. Claribel Lucy Cruz Aguila Proceso de pruebas de aceptación para un software de gestión. 2007.
12. Meléndez, C. "Plan de pruebas de aceptación", 2005.
13. Quesada, J.A.L., Ingeniería del Software. Pruebas del software. 2007.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

1. Humphrey, W., "Introducción al Proceso de Software Personal". 2001.
2. ISO 9000:2000, "Sistema de Gestión de la Calidad Principios y Vocabulario", 2000.
3. Pressman Roger, "Ingeniería de Software: Un enfoque práctico", Quinta edición, McGraw-Hill, 2002.
4. ISO/IEC 12207:1995, Norma Internacional, Ciclo de Vida del Software.
5. Microsoft Corporation, "Microsoft Internet Data Center", Capítulo 9, Diseño del proceso de prueba, 2001.
6. Cuatrecasas, Luis, "Gestión Integral de la Calidad"; Gestión 2000, Barcelona, 2001.
7. Piattini, Mario, García Félix O, "Calidad en el desarrollo y mantenimiento del software", RA-MA Editorial, Madrid, 2003.
8. Cockburn, Alistair, "Writing effective use cases", Addison Wesley, 2000.
9. Revista Software Gurú, Artículo "El Proceso de la Prueba de Software. Lenguaje de Definición de Proceso" Año 01, No. 06. Noviembre-Diciembre 2005.
10. RUP, "Ayuda de Rational Unified Process", 2003.
11. González, J. "Las normas de la Calidad del Software". España, 2002.
12. Myers, G. J. "The art of Software Testing", 2004.
13. Notenboom, B. B. A. E. "Testing Embedded Software", 2003.
14. Scalone, L. F. "Estudio Comparativo de los Modelos y Estándares de Calidad del Software". Buenos Aires, Universidad Tecnológica Nacional, 2006.
15. Juristo, N., Moreno, A. M., and Vegas, S. "Técnicas de evaluación de software", 2006.
16. "IEEE 829: Estándar para la Documentación de las Pruebas de Software", 1998.
17. "ISO 12207: Estándar para el Ciclo de Vida del Software", 1996.
18. Ruiz, J. Z. "Diagnostico Organizacional para la Implantación de los Modelos de Calidad en Pequeñas Empresas de Software", 2006.
19. Meléndez, C. "Plan de pruebas de aceptación", 2005.
20. Quesada, J. A. L. "Ingeniería del Software. Pruebas del software.", 2007.

BIBLIOGRAFÍA

21. MSc. Yamilis Fernández, I.D.A.M., Ing. Violena Hernández Aguilar, Ing. Michael González Jorrín, Ing. Pérez, Ing. Rosalía Lucia Cué Delgado, Lic. Claribel Lucy Cruz Aguila. "Proceso de pruebas de aceptación para un software de gestión. 2007.
22. Valencia R., G. A. "Subcontratación del servicio de pruebas de software", 2007.
23. Carmentales, Nadiesda Sanz y Rivas, Geiser Arcio Pérez. "Pruebas de Aceptación Parciales del Cliente". Ciudad de La Habana, Cuba: Universidad de las Ciencias Informáticas (UCI), 2007. Tesis de Diploma
24. Alonso, Yosbel Ernesto González y Arribas, Yoan Manuel Cabrera. "Proceso de Pruebas del Registro de Áreas de Salud de la Atención Primaria del Sistema de Información para la Salud". Ciudad de La Habana, Cuba: Universidad de las Ciencias Informáticas (UCI), 2007. Tesis de Diploma.
25. Gálvez, Annerys Aguiar y Reyes, Alain Bedoya. "Propuesta de técnicas de prueba de software para elevar la calidad en proyectos productivos de Entornos Virtuales". Ciudad de La Habana, Cuba: Universidad de las Ciencias Informáticas (UCI), 2007. Tesis de Diploma.
26. Palma, Yuniet del Carmen Toll y Torres, Yilennis Mendoza. "Propuesta de manual de procedimiento de Pruebas de Sistema y su aplicación en el Proyecto CICPC". Ciudad de La Habana, Cuba: Universidad de las Ciencias Informáticas (UCI), 2007. Tesis de Diploma.
27. Martínez, Zamira Segoviano y Lores, Yusmila Vidiaux. "La reutilización como parte de la calidad del software. Su aplicación en la UCI". Ciudad de La Habana, Cuba: Universidad de las Ciencias Informáticas (UCI), 2007. Tesis de Diploma.

ANEXOS

Anexo 1: Entrevista realizada a miembros de calidad UCI.

- 1- ¿Qué tipo de personal es el encargado de realizar las pruebas de aceptación?
- 2- ¿Se guían por alguna metodología para realizarlas?
- 3- ¿Existe alguna documentación en la UCI referente a estas pruebas?
- 4- ¿Cuáles son las principales dificultades que existen a la hora de realizarlas?
- 5- ¿Se le da la debida importancia a este proceso?
- 6- ¿Si se encuentra no conformidades como se procede arreglarlas?
- 7- ¿Cuáles son los artefactos que se generan en las pruebas de aceptación?
- 8- ¿Cuáles son los roles que intervienen en estas pruebas?

Anexo 2: Normas y Estándares.

Entre algunas de las normas mayoritariamente utilizadas para certificar los procesos de validación se encuentran: la ISO/IEC 12207 presenta los procesos en el ciclo de vida del software, entre ellos se encuentra el proceso de validación donde se desarrollan todas las actividades para determinar si un sistema determinado cumple con las especificaciones y requisitos para los cuales fue realizado, la IEEE 829 muestra como se deben elaborar un conjunto de los artefactos generados en el proceso de pruebas y la IEEE 1062 define un conjunto de actividades a realizar en el caso de la adquisición de un sistema informático, entre ellas, se encuentran la aceptación de software.

ISO/IEC 12207: 1995

En la norma ISO/IEC 12207: presenta los procesos fundamentales en el ciclo de vida del software empleados para adquirir, suministrar, desarrollar, operar y mantener productos de software. Su objetivo es presentar vías para los usuarios conocedores de esta norma, se orienten por sí mismos y sepan aplicarla juiciosamente según sus objetivos y propósitos.

El propósito del proceso de validación es determinar si un sistema ya construido cumple con las especificaciones y requerimientos para los cuales fue realizado.

ANEXOS

Este proceso puede ser ejecutado a distintos niveles de independencia. El nivel de independencia puede extenderse desde una persona o grupo de personas de la misma organización hasta una persona o grupo de personas de una organización diferente con distintos niveles de separación. En el caso donde el proceso sea ejecutado por una organización independiente del proveedor, desarrollador, operador, o de mantenimiento, es llamado Proceso de Validación Independiente. (ISO, 12207)

Un proceso esta compuesto por actividades, a su vez las actividades por tareas y las tareas son actividades específicas que contribuyen al cumplimiento de la misión general u otros requerimientos.

Una de las actividades que muestra la norma, es la actividad de validación.

Proceso de validación. Esta actividad esta compuesta por las siguientes tareas (ISO, 12207):

Preparar la selección de los requerimientos de pruebas, casos de prueba, y las especificaciones de pruebas para analizar los resultados de las pruebas.

Asegurarte de que éstos prueben los requisitos, casos de la prueba, y las especificaciones de prueba refleja los requisitos particulares para el uso previsto específico.

Conducir las pruebas en subclases, incluye:

Probando con la tensión, el límite, y las entradas singulares.

Probando el producto de software por su capacidad de aislar y de minimizar el efecto de los errores; es decir, respuesta elegante sobre la falta, pedido por el asistente de operaciones sobre la tensión, límite, y condiciones singulares.

Probar los usuarios representativos puede alcanzar exitosamente sus tareas previstas usando el producto de software.

Validar que el producto de software satisface su uso previsto.

Probar el producto de software como apropiado en las áreas seleccionadas del ambiente seleccionado.

La norma plantea las actividades a desarrollar en el proceso de validación de forma general sin especificar detalladamente las tareas a realizar en cada actividad. Por tanto, se opina, se puede utilizar como base para adquirir conocimientos generales en caso de desconocimiento de las posibles actividades a realizar, por el contrario, si se tiene una idea clara del logro a alcanzar en el proceso de validación, se considera no sea conveniente consultar este estándar.

IEEE 829: 1998

La norma IEEE 829-1998 Estándar para la Documentación de la Prueba de Software es un estándar donde se especifica como elaborar la documentación de un plan de pruebas.

Dentro de los documentos de encuentra el Plan de Pruebas, Especificación del Diseño de Pruebas, Especificación de Casos de Prueba, Especificación de Procedimiento de Prueba, Reporte de Transmisión de elementos de Prueba, Registro de Pruebas, Reportes de Incidentes de Pruebas, Reporte de Resumen de Pruebas. (IEEE, 829)

De toda la documentación generada en el proceso de pruebas, una de las más importantes es el plan de pruebas, de los artefactos que la IEEE 829 propone, es hacia ese punto al que se dirige la atención.

El propósito del plan de pruebas incluye determinar el alcance, la estrategia, los recursos y la organización de las actividades de pruebas. Además, se identifican los elementos de inicio de prueba, la forma en que serán probados, las tareas que serán ejecutadas, el personal responsable para cada tarea y los riesgos asociados con este plan (IEEE, 829).

En la norma se establece una estructura de este plan de pruebas, entre los elementos encontrados se puede citar: el identificador del plan, la introducción, elemento de prueba, aspectos que serán probados, aspectos que no serán probados, la metodología, criterio de fallo, criterio de suspensión y requerimientos, prueba entregable, probando tareas, ambiente necesario, responsabilidades, capacitación, cronograma, riesgos y contingencias, y aceptación (IEEE, 829). Cada uno de los aspectos es detallado en la norma, explicando que elementos específicos deberán ser tratados en cada sesión.

También se especifican las referencias a otros documentos de importancia, por lo general entregables para el cliente, como por ejemplo: Plan de Revisiones, Plan de Aseguramiento de la Calidad o Plan de Proyecto. Estas referencias son importantes porque representan los enlaces existentes entre las diferentes actividades involucradas en el proceso de pruebas, como lo es la calidad, por citar un ejemplo.

Los documentos generados en el proceso de pruebas del software deben estar de acorde con los objetivos trazados al inicio de la planificación de las mismas, algunos pueden ser omitidos y pueden aparecer otros totalmente nuevos, esto dependerá de las condiciones específicas del proyecto y las necesidades que tenga el equipo de desarrollo en conjunto con el cliente.

ANEXOS

La realización del plan de pruebas de aceptación se rige por algunos aspectos planteados en esta norma.

Anexo 3: Pruebas de aceptación parcial y final.

Diagramas.

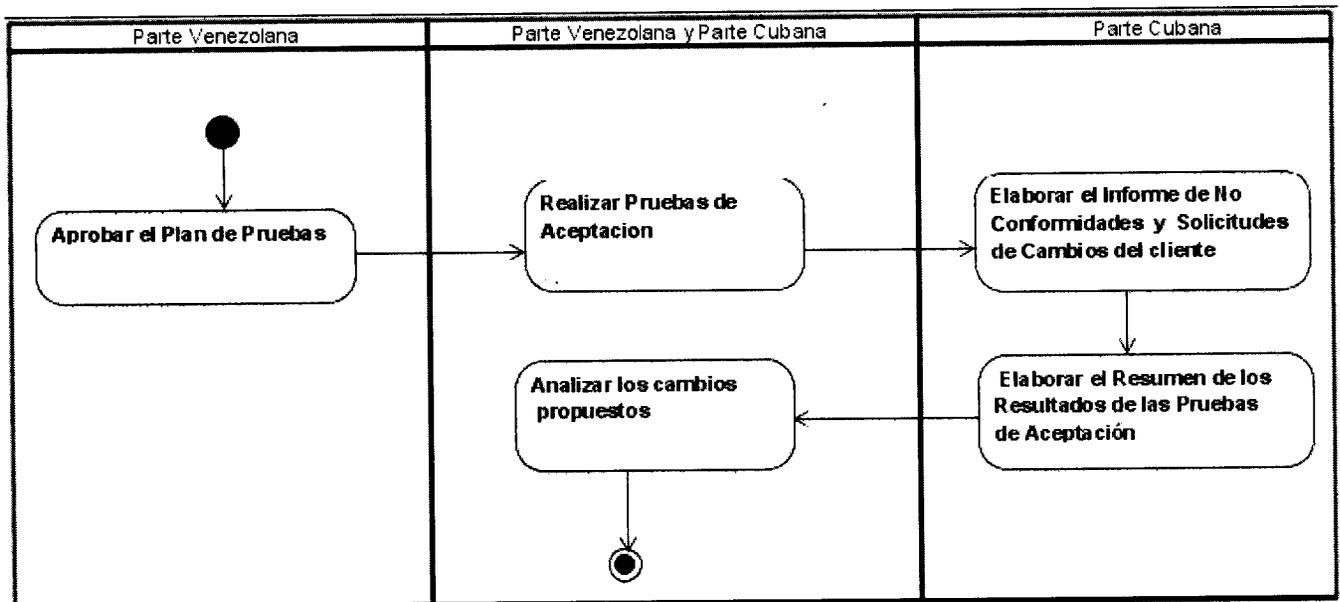


Figura 1: Flujo de trabajo de las pruebas de aceptación parcial del cliente.

ANEXOS

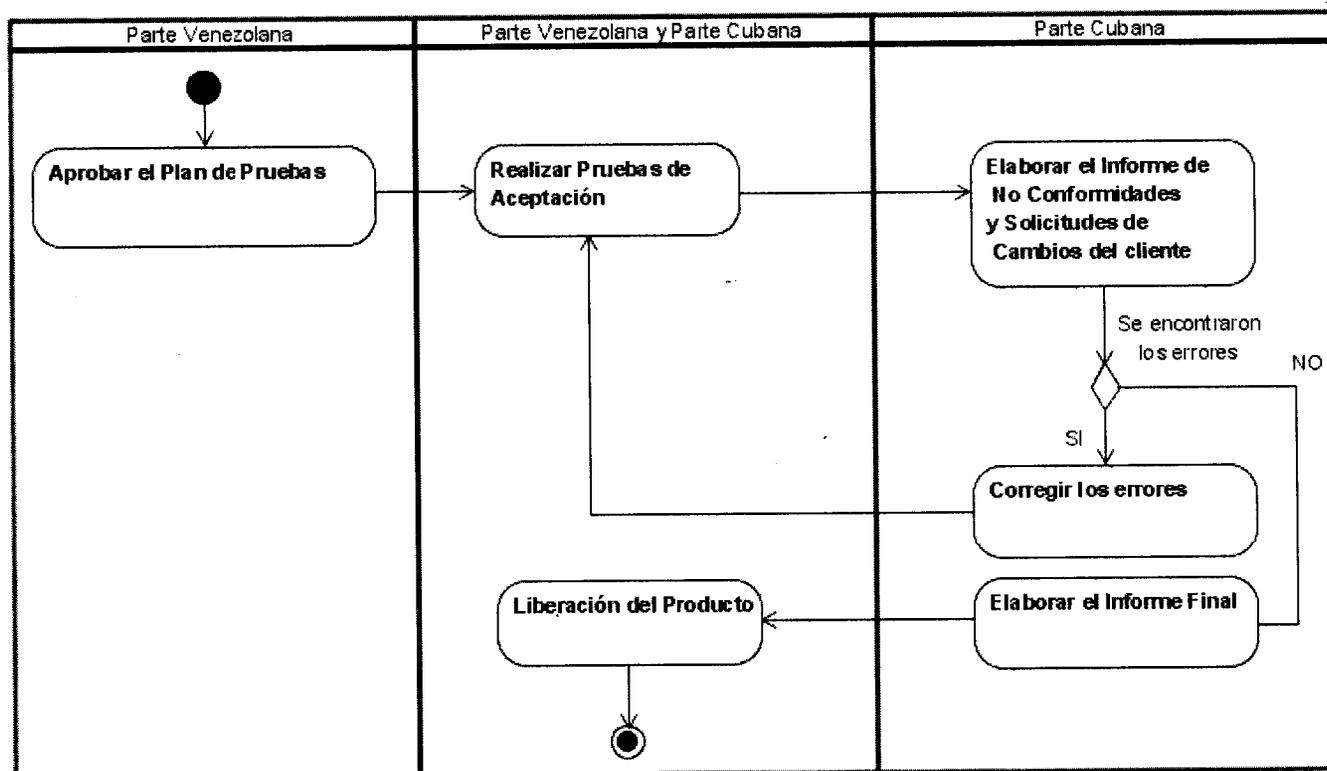


Figura 2: Flujo de trabajo de las pruebas de aceptación final del cliente.

Anexo 4: Diseño de casos de uso.

La descripción de casos de usos cada diseñador del negocio tiene sus reglas para realizarlas. Desde el punto de vista de la información necesaria para la elaboración del caso de pruebas el aspecto más importante es el Flujo de Eventos el cual consta de dos partes el flujo básico de eventos y los flujos alternativos de eventos. El flujo básico de eventos debe cubrir lo que sucede "normalmente" cuando se realiza el caso de uso. Los flujos alternativos de eventos cubren el comportamiento de una opción o excepción relativa al comportamiento normal, además de variaciones de este. Se puede pensar en los flujos alternativos como "desvíos" del flujo básico de eventos.

Nombre del Caso de Uso	Gestionar Existencia de Libros
------------------------	--------------------------------

ANEXOS

Actores	Administrador.
Propósito	Permitir registrar, eliminar y modificar datos acerca de los Libros.
Resumen	Es aquí donde se registran, eliminan, y se modifican los datos de los Libros que se entran en el almacén.
Precondiciones	<ul style="list-style-type: none"> • Administrador del sistema ya autenticado.
Postcondiciones	<ul style="list-style-type: none"> • Se registra un medio básico, se actualizan los datos o se elimina el mismo.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El administrador del sistema necesita registrar, eliminar y modificar los datos de un libro.	<p>El sistema ejecuta alguna de las siguientes acciones:</p> <ul style="list-style-type: none"> a) Si el administrador decide registrar un medio básico, ir a la sección "Registrar Libro" b) Si el administrador decide actualizar las características de un libro, ir a la sección "Modificar Datos". c) Si el administrador decide eliminar un libro, ir a la sección "Eliminar Libro".
Sección "Registrar Libro"	
Acciones del Actor	Respuesta del Sistema
2. El Administrador del Sistema entra los datos del Libro para realizar su registro en la aplicación.	<p>2.1 El sistema verifica que los campos del Código, el Nombre y la cantidad estén llenos.</p> <p>2.2 El sistema verifica que este Libro no exista.</p> <p>2.3 El Libro se almacena en el sistema.</p> <p>2.4 El sistema muestra un mensaje informándosele al administrador que ya ha sido efectuado el registro del libro y finaliza el caso de uso.</p>

ANEXOS

Curso alternativo	
	<p>2.1 El sistema emite un mensaje para que llene los campos obligatorios.</p> <p>2.3 Si el libro existe El sistema muestra un mensaje informativo y finaliza el caso de uso.</p>
Sección "Actualizar Datos"	
Acciones del Actor	Respuesta del Sistema
2. El administrador del sistema selecciona el libro a modificar.	2.1 El sistema brinda la posibilidad de modificar los datos.
3. El administrador del sistema realiza las actualizaciones deseadas.	<p>3.1 El sistema verifica que los campos obligatorios estén llenos.</p> <p>3.2 El sistema actualiza la Información y finaliza el caso de uso.</p>
Curso alternativo	
	3.1 El sistema emite un mensaje para que llene los campos obligatorios.
Sección "Eliminar Libro"	
Acciones del Actor	Respuesta del Sistema
2. El administrador del sistema selecciona el libro a eliminar.	<p>2.1 El sistema pide confirmación.</p> <p>2.2 El sistema elimina el Libro.</p>
Curso alternativo	
	2.1 Si no se da la confirmación El sistema no se elimina el libro

Tabla 1: Descripción de un caso de uso.

ANEXOS

Anexo 5: Pasos para diseñar casos de pruebas.

Para cada caso del uso, generar un sistema completo de escenarios.

Lo primero que se hace es leer la descripción textual del caso de uso e identificar todas las combinaciones posibles de caminos de ejecución del caso de uso, es decir todas las combinaciones entre el camino principal y los caminos alternativos y se le asigna un nombre. Cada combinación será un escenario de uso. Esta información se representa en una matriz, que se denomina Matriz Parcial de escenarios, donde se tiene en cuenta el nombre del escenario, el flujo donde comienza y si contiene algún flujo alternativo.

Nombre del escenario	Flujo donde empieza	Alternativo
Escenario 1 "Registro exitoso"	Flujo Básico	
Escenario 2 "Faltan datos obligatorios"	Flujo Básico	A1
Escenario 3 "Ya existe el libro"	Flujo Básico	A2

Tabla 2: Matriz Parcial de escenarios.

Identificar los Casos de Prueba.

Luego de identificados los diferentes escenarios de prueba, se identifican las variables (la información necesaria para que el escenario se ejecute) que hacen que dicho escenario se comporte de una forma o de otra, es decir como un flujo básico o alternativo y nuevamente esta información se representa en una matriz, llamada Matriz de Casos de Prueba, donde se representa además el comportamiento esperado del sistema.

Id	del	Escenario	Códig	Nombre	Cantida	Respuesta del Sistema
----	-----	-----------	-------	--------	---------	-----------------------

ANEXOS

escenario		o		d	
EC 1	Escenario 1: "Registro exitoso"	V	V	V	El sistema muestra un mensaje informándole al administrador que ya ha sido efectuado el registro del libro.
EC 2	Escenario 2: "Faltan datos obligatorios"	Alguno de los datos toma valor l.			El sistema emite un mensaje para que llene los campos obligatorios
EC 3	Escenario 3: "Ya existe el libro"	V	V	V	El sistema muestra un mensaje informativo.

Tabla 3: Matriz de Casos de Prueba.

Identificar Valores de Datos para las Pruebas.

Una vez que todos los casos de prueba se han identificado, todos ellos deben repasarse y validarse para asegurar exactitud e identificar los casos de prueba redundantes o que faltan. Entonces, una vez que sean aprobados, el paso final es sustituir los valores reales de los datos, porque la idea es probar con datos reales.

Id del escenario	Escenario	Código	Nombre	Cantidad	Respuesta del Sistema	Resultado de la prueba
EC 1	Escenario 1: "Registro exitoso"	BN2345	El código Davinci	20	El sistema muestra un mensaje informándole al administrador que ya ha sido efectuado el registro del libro.	Se especifica que sucedió al realizar la prueba, pueden existir dos variantes: que la respuesta del sistema sea la esperada o que no lo sea.
EC 2	Escenario 2: "Faltan datos obligatorios"	BN4567	" "	" "	El sistema emite un mensaje para que llene los campos obligatorios	
EC 3	Escenario 3: "Ya existe el libro"	TY345	El amor en los tiempos del cólera	5	El sistema muestra un mensaje informativo.	

ANEXOS

Tabla 4: Matriz de Casos de Prueba con los valores de los datos.

Anexo 6: Formato para realizar el plan de pruebas utilizando la prueba de caja negra.

Código de la Prueba	Fecha	Nombre de la interfaz	Código de clases de equivalencia	Responsable de realizar la prueba

Tabla 5: Formato para el plan de pruebas.

Donde:

- Código de la prueba: Este es el código que el responsable de la prueba debe asignarle.
- Fecha: Fecha en la cual se debe realizar la prueba.
- Método de prueba: El responsable tiene que escribir que método de prueba va realizar.
- Nombre de la Interfaz: Indica el nombre de la interfaz a probar.
- Responsable de realizar la prueba: El nombre del responsable que tiene que realizar la prueba.
- Número del grupo de clases de equivalencia: Se especifica el grupo de clase de equivalencia al que pertenece.

Anexo 7: Formato de para presentar los resultados de las pruebas

Fecha (dd/mm/aa)	Código de la prueba	Detalles del caso de prueba	Resultados esperados	Resultados Obtenidos	Fecha Final y Aprobación

Tabla 6: Formato para presentar los resultados de las pruebas.

ANEXOS

Donde:

- Fecha indica la fecha en que se realizó la prueba
- Código de la prueba indica el código de las diferentes pruebas hechas a los métodos de la responsabilidad.
- Detalles del caso de prueba indica una breve descripción de la prueba a realizar.
- Resultados esperados indica los resultados que se esperan obtener antes de realizar la prueba.
- Resultados obtenidos son los resultados que se obtuvieron al realizar la prueba.
- Fecha Final y resultados indican la fecha en que se culminaron las pruebas y los resultados de esta, es decir, si el método está listo o queda pendiente.

Anexo 8: Descripción de los Casos de pruebas

[Lugar donde se realizan las pruebas]

[Casos de Pruebas.]

[Nombre del Proyecto]

[Nombre del módulo]

[Nombre del Caso de uso que cubre el caso de prueba]

1- Descripción General

[Se describe de forma general las pruebas que se ejecutaran al caso a probar, así como se describe brevemente el caso de uso]

[Las pruebas realizadas a este caso de uso son:]

Prueba 1

Prueba 2

Prueba n

1.1 CPR 1: Prueba 1

1.2 Descripción.

[Se describe brevemente como se debe ejecutar el caso de prueba]

1.3 Flujo Central.

ANEXOS

[Se debe describir paso por paso el como se debe ejecutar el caso de prueba, con el fin de que el encargado de ejecutarla la realice sin problemas y esta pueda cumplir el objetivo para la cual fue creada. Se recomienda que se describa en forma de viñetas]

1.4 Condiciones de Ejecución

[Se describen todos los elementos que deben estar presentes para lograr una correcta ejecución del caso de prueba.]

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la prueba	Observaciones	Cumplimiento %
[Se describen las clases válidas para la prueba]	[Se describen las clases no válidas para la prueba]	[Se describen los resultados que se sabe que debe devolver el caso de prueba]	[Se describe como se ejecutó la prueba si fue correcta o no]	[Se describe las observaciones que sean pertinentes]	[Se describe el cumplimiento de la prueba en %]

Tabla 7: Descripción de los casos de pruebas.

Anexo 9: Registro de defectos y dificultades

Elemento	Nº	Descripción de la No conformidad	Aspecto correspondiente	Etapas de la detección	Importante	Recomendaciones
[Se describe la naturaleza del error]	[número]	[Se describe la no conformidad]	[Se describe el error]			

Tabla 8: Registro de No Conformidades.

[Nombre del que diseño la prueba]

[Fecha]

[Nombre del que ejecutó la prueba]

[Fecha]

GLOSARIO DE TÉRMINOS

GLOSARIO DE TÉRMINOS

- **Calidad:** Satisfacción de las necesidades del cliente.
- **Cliente:** Persona, organización o grupo de personas que se encargan de la construcción de un producto de software.
- **Proceso:** Secuencia de actividades que tienen un marcado inicio y fin.
- **Lista de Chequeo:** Una serie de aspectos que se utilizan para comprobar un producto o proceso.
- **Pruebas de Software:** Son el proceso de ejecución de un programa con la intención de descubrir un error.
- **Pruebas de aceptación:** Son las pruebas realizadas por el cliente para validar el software.
- **Caso de Uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos.
- **Caso de Prueba:** Especificación de cómo probar el sistema, incluyendo la entrada, salida y condiciones bajo las cuales se prueba.
- **No Conformidad:** Un error detectado en una funcionalidad que está pactado con el cliente.
- **Solicitud de Cambio:** Sustitución o transformación que se realiza a un elemento de configuración.
- **Validación:** No es más que verificar que un producto determinado cumple con los requisitos que fueron pactados con el cliente.
- **Métrica:** Es el término que describe muchos y muy variados casos de medición.