

Universidad de las Ciencias Informáticas

Facultad 1



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

Título: Propuesta de procedimiento general para pruebas de liberación de los productos de la Facultad 1.

Autores: Noralis Aldana La 'O.

Jorge Luis Ramos Medina.

Tutoras: Lic. Regla Silva Calderón

Ing. Jenny Suarez Vázquez

Ciudad de La Habana, Junio de 2009

“Año del 50 aniversario del triunfo de la Revolución”

Declaración de Autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Noralis Aldana La O

Jorge Luis Ramos Medina

Firma del Autor

Firma del Autor

Lic. Regla Silva Calderón

Ing. Jenny Suarez Vázquez

Firma del Tutor

Firma del Tutor



Dedicatoria



A mi madre, por ser la máxima responsable de mi existencia, por amarme con tanta pasión y dedicación y por ser ella, el motor impulsor para la realización de este sueño. “Verme graduada”

A mi primita Naika, por ser la luz de mi vida, la hermana que dios no me dio y por darme fuerzas cuando yo pensé que no podía más. “Te quiero mucho mi niña”

A mi abuelito Pipo, porque aunque ya no forme parte de este mundo, en el mío si lo está, y sé, que para él era un orgullo verme graduada.

Noralis

A mi madre, por ser mi mamá y mi papá, que tanto se esforzó porque yo llegase hasta aquí, por ser mi luz, mi guía, al inicio y el final de mi carrera, el mayor motivo de todo esto.

A mi único hermano Migue y a mi sobrinito hermoso .A mi familia en general, en especial a mis tías que más quiero Yolanda y Silvina.

A mis amigos –hermanos Yadir, Yisel y Yannelys, amigos como los que ya no existen en el Mundo, y de los que estoy orgulloso. A todos mis compañeros muchísimas gracias por todo, y sobre todo también a mi Virgencita de la Caridad del Cobre, a mi Dios y a todos mis santos.

Jorge Luis





Agradecimientos



Hemos pasado cinco años de nuestras vidas en la universidad, donde nos hemos formado para ser unos excelentes profesionales. Hoy damos fin, con este trabajo a este largo viaje, lleno de sacrificio, abnegación y entrega y de esta forma se sentarán las bases para el comienzo de otra etapa importante en nuestras vidas.

Primero, queremos agradecer a la Revolución y a Fidel Castro, por ser el máximo responsable de la creación de este centro de estudio.

A la Universidad de las Ciencias Informáticas por habernos dado la oportunidad de finalmente cumplir nuestro sueño y el de nuestras familias y de conocer tantos amigos, con los cuales compartimos momentos importantes de nuestras vidas.

A todas las personas que han estado a lo largo del camino a nuestro lado, en las buenas y en las malas

A nuestra tutora Jenny, porque supo confiar en nosotros, diciéndonos día a día que si se podía.


A nuestra tutora Reglita, por dedicar tanto al desarrollo de esta investigación y brindarnos toda la información necesaria para la realización de la misma.

A Carlos Luis Milian, por dedicar parte de su preciado tiempo en uno de los momentos más importantes de la carrera.


A todos queremos ofrecerles nuestros más sinceros agradecimientos por su apoyo, dedicación, y por hacer posible este sueño.

Noralis y Jorge Luis





Agradecimientos



A mi tía Yonaika, por quererme como a una hija, apoyarme en todo, ser incondicional conmigo y recordarme cada segundo de su vida lo importante que soy en la suya.

“Para ti todo mi amor, mi tercera madre”

A mi hermano Noslem, por adorarme con tanta pasión, sentir tanto orgullo de mi existencia y darme todas las fuerzas del mundo, cuando yo pensé que todos los caminos estaban cerrados. “Gracias por existir”

A mi abuela Mora, por ser mi segunda madre y quererme como tal.

A mi tía Idalsis, porque aunque muy lejos este de mí, sé que me quiere más que a nada en este mundo. “Te extraño mucho mi titi”

A mi abuelita Mima, por ser tan dócil y comprensiva conmigo y recordarme tantas veces, que me quiere mucho. “Eres especial”

A mi tía Teri, por ser yo su sobrina preferida y demostrármelo en cada segundo de su vida.

A mi papá, por haberme dado el privilegio de existir.

A mis primos, por tenerme como un ejemplo a seguir. También a mi sobrino Carlito y mi hermanito Alfredito.

A mi familia por vivir conmigo este sueño.

A Yane, Katy y Benilde, por ser desde sus comienzos verdaderos amigos

A Mirtha, por quererme como una nieta y ayudarme tanto, todos estos años de mi carrera.

A Alejandro, por apoyarme siempre en todo y saber ser amigo y compañero.

A Ailín, porque aunque en momentos de la vida, no nos supimos entender, me demostró que si sabía ser amiga.

A Alina, por hacerme parte de su club de amigos y escogerme entre ellos como una de sus preferidas.


A Yunaldis, porque a pesar de haber sido la última en llegar a este club de amigos se ha sabido ganar de una forma muy especial ese lugar.

A Leidis, Alien, Daimaris, Chanel, Yitsy, Yisel, Olivier, Guille, Arian, Charly, Marianny, Raúl, Alexander Bonet, Elisa, Gustavo, Yusmara, Damián, a todos ellos gracias por aceptar mi amistad incondicionalmente.


A todos una y mil veces GRACIAS, por hacer realidad mi sueño

Noralis.





Agradecimientos



Quiero agradecerle sobre todas las cosas a mi mamita linda y bella, por ser mi mamá y mi papá, por ser mi luz, mi guía, al inicio y al final de mi carrera, el mayor motivo de todo esto, por querernos a mi hermano Migue, a mi sobrinito Cristian Alejandro y a mí, por ser su mundo. Más que todo por ser su hijo y ella ser única.

A mi hermano Migue, este camino que he transitado es para que veas a tu coquico graduado de la Universidad, si yo lo estoy tu también lo estas mi hermanito.

A mi familia en general, en especial a mis tías Silvina y Yolanda que siempre han estado ahí apoyándome.

A mis amigos –hermanos Yadir, Yisel Filiu y Yannelys, amigos como los que ya no existen en el Mundo, de los que estoy muy orgulloso por aceptarme en sus corazones, a estos los quiero un chorro.

A mis amigos del pre-universitario Emilito, Yiliana, Leivis, Lilita, Yile, Yisel Salvador, Rachel, Maryant, los adoro.

A los que recorrieron el mismo camino conmigo, la Yudi, Lily, Osmany, la Zuny, Suri, Yahima, Arianna, Marlen y a mis nuevos amigos José, Yuniesky.

A todos mis compañeros muchísimas gracias por todo.

A mi madre de la UCI Laurita, gracias y mil veces gracias, los motivos son muchos, más que todo por tu cariño y tu amor incondicional.

A las tías del comedor María y Lachi, a las del docente Yoyi y María, y a todas las de beca.

Y más que a nadie, a ti mi Virgencita de la Caridad del Cobre, a mi Dios y a todos mis santos queridos que estuvieron y están conmigo.

Jorge Luis



Resumen

La producción de la facultad 1 de la Universidad de las Ciencias Informáticas se encamina hacia el desarrollo de tecnologías web para la digitalización de la sociedad, y para la exportación, para ello se encuentra organizada en dos centros que agrupan a diferentes proyectos. Una vez terminado el producto, antes de comenzar su despliegue, se le debe realizar diferentes tipos de pruebas, para garantizar que salga con la calidad requerida y cumpla con los requisitos del cliente. Sin embargo, en la facultad 1 no se han definido actividades obligatorias para realizarle pruebas de liberación al producto.

La investigación tiene como objetivo proponer un procedimiento general para pruebas de liberación de los productos de la facultad 1. Para ello se enfoca en la planificación, desarrollo y control de las pruebas de liberación de los productos desarrollados en los centros de la facultad 1, esto posibilitará, que el producto final, una vez revisado, cumpla todos los requisitos establecidos y posea un mejor índice de calidad. Se analiza la situación actual con el fin de desarrollar una propuesta aplicable a los dos centros de la facultad 1.

El diseño de este procedimiento está basado en las especificaciones de la NC ISO/IEC 9126-1:2005 así como en el ciclo Deming para la mejora continua. La propuesta permitirá que las aplicaciones web desarrolladas en estos centros, lleguen al Laboratorio Industrial de Pruebas (LIPS) con la menor cantidad de defectos posibles y cumplan de esta forma con los requisitos establecidos por el cliente.

Palabras Claves: procedimiento, prueba, liberación.

| | |
|---|-------------------------------------|
| Introducción..... | 1 |
| CAPÍTULO 1..... | 6 |
| Fundamentación teórica | 6 |
| 1.1..... | Marco Conceptual |
| | 6 |
| 1.2 Calidad del software | 8 |
| 1.3 Aseguramiento de la calidad..... | 9 |
| 1.4 Gestión de la calidad | 9 |
| 1.5 Control de la calidad | 11 |
| 1.6 Sistema de Calidad..... | 11 |
| 1.7 Ciclo Deming..... | 11 |
| 1.8 Mejora continua | 12 |
| 1.9 Pruebas de software | 12 |
| 1.9.1 Pruebas de liberación | 18 |
| 1.9.2 Método de prueba | 19 |
| 1.10 Modelos de calidad..... | 21 |
| 1.10.1 Importancia de la evaluación y medición de la calidad del software ... | 21 |
| 1.10.2 Normas ISO | 22 |
| 1.11 Experiencias en la evaluación de los productos en los centros | 28 |
| 1.12 Análisis de causas y efectos en la preparación de un producto en la facultad 1 aplicando el método de Ishikawa | 28 |
| 1.13 Conclusiones Parciales..... | 30 |
| CAPÍTULO 2..... | 32 |
| 2.1 Propuesta del procedimiento de pruebas de liberación para los centros de la facultad 1 | 33 |
| 2.2 Conclusiones Parciales | 58 |
| CAPÍTULO 3..... | 59 |
| Validación del procedimiento propuesto | 59 |
| 3.1. Introducción..... | 59 |
| 3.2 Proceso de selección de especialistas..... | 59 |
| 3.2.1. Determinar la cantidad de especialistas y conformar el listado de especialistas | 60 |
| 3.2.2 Confirmar la participación de especialistas | 61 |
| 3.3 .Elaboración de la encuesta | 61 |
| 3.4. Resultados de la evaluación..... | 61 |
| 3.5 Resultados de las encuestas realizadas a los especialistas..... | 63 |
| 3.6.Beneficios de la propuesta..... | 66 |
| 3.7.Conclusiones Parciales..... | 66 |
| Conclusiones | 67 |
| Recomendaciones | 68 |
| Referencias bibliográficas | 69 |
| Bibliografía | 72 |
| Anexos..... | Error! Bookmark not defined. |
| Anexo 1: Organigrama | Error! Bookmark not defined. |
| Anexo 2: Formulario de solicitud de Revisión de producto de software .. | Error! Bookmark not defined. |
| Anexo 4: Parte 1: Modelo de la calidad NC-ISO/IEC9126-1: 2005..... | Error! Bookmark not defined. |

| | |
|--|-------------------------------------|
| Anexo 5: Informe de los Resultados de la Inspección del Producto | Error! Bookmark not defined. |
| Anexo 6: Acta de Entrega | Error! Bookmark not defined. |
| Anexo 7: Acta de Aceptación | Error! Bookmark not defined. |
| Anexo 8: Plantilla de Diseño de casos de Pruebas..... | Error! Bookmark not defined. |
| Anexo 9: Plantilla de No conformidades | Error! Bookmark not defined. |
| Anexo 10: Minuta de reuniones..... | Error! Bookmark not defined. |
| Anexo 11: Plantilla de evaluación de los pesos..... | Error! Bookmark not defined. |
| Anexo 12: Informe de los resultados de la evaluación ... | Error! Bookmark not defined. |
| Anexo 13: Encuesta..... | Error! Bookmark not defined. |
| Glosario de Términos..... | 75 |

Introducción

Las pruebas de liberación de software no son más que una actividad de evaluación del producto. La experiencia ha demostrado que no se deben realizar al final del desarrollo del producto, sino durante todo su ciclo de vida. Estas no tienen como objetivo destruir el producto, sino aportar información sobre el mismo para tomar decisiones y acciones correctivas.

En muchos países de Europa y otras partes del mundo, han recurrido a la vía de pruebas de liberación de software, con el propósito de generar un espacio en el cual se puedan confrontar y poner a prueba el desempeño de productos y servicios, contribuyendo, en el desarrollo y la ejecución de proyectos de investigación, en los campos afines a su misión en los ámbitos académico e industrial. Muchos de estos países cumplen con la asesoría a las grandes empresas en la selección y programación de las pruebas necesarias. Esto se hace con el objetivo de asegurar el buen desempeño de los productos de la empresa, desde los aspectos de usabilidad durante su utilización, pruebas de productos y sistemas con la participación de usuarios reales.

A partir de la necesidad de controlar y comprobar la calidad de los productos, se comienza a idear un sistema que permitiera detectar los defectos del producto antes de lanzarlo al mercado. Como respuesta a esta búsqueda de mejora y control de los errores de los productos, se comienzan a ejecutar a nivel mundial, pruebas de liberación de los productos, con el propósito de ofrecer a los clientes, mejores servicios y recursos, tanto para los proyectos que se llevan a cabo en las oficinas, como para los que se realizan en instalaciones externas. (AENOR, 2008)

Cuba ha dado grandes pasos para entrar en el mercado mundial con sus productos. A pesar de no representar una gran potencia tecnológica, en los últimos años, se ha venido trabajando, por elevar el nivel de las Tecnologías de la Información y las Comunicaciones (TIC), que ocupan un papel primordial en la sociedad y la economía de cualquier nación.

En la nación cubana, se encuentra el Ministerio de Informática y las Comunicaciones (MIC). Como parte de la búsqueda de lograr, que los productos cubanos alcancen un alto nivel de reconocimiento en el mercado mundial, y que tenga la calidad necesaria, el mismo se apoya en la realización de pruebas y emplea modelos de calidad reconocidos internacionalmente. La

Introducción

Universidad de las Ciencias Informáticas (UCI) juega un rol muy importante en el desarrollo tecnológico de Cuba, como protagonista número uno de los papeles principales en el desarrollo de software y por consiguiente en garantizar la calidad de los mismos.

Para liberar un producto muchas facultades lo hacen después que el producto está terminado, mayormente realizan pruebas de caja negra y estas se organizan en laboratorios de pruebas dentro de la facultad, otras también hacen pilotos en el entorno natural del cliente antes del despliegue. Con este objetivo la UCI ha creado diversos grupos de proyectos de calidad que incursionan en este campo y se ocupan tanto de la producción como de la calidad de los productos de software que salen de los mismos.

Existe además el Laboratorio Industrial de Pruebas (LIPS), que asesora a cada facultad y recibe los productos de sus centros para una posterior revisión y realización de pruebas de liberación más profundas de los productos de exportación que se desarrollan en la universidad, como son las aplicaciones Desktop y Web. Este laboratorio se encarga de encontrar al producto la mayor cantidad de no conformidades posibles, para luego ser liberado.

La facultad 1 está compuesta por dos centros: el Centro de Identificación y Seguridad Digital (CISED) y el Centro de Informatización Universitaria (CENIA). Estos centros tienen internamente varios proyectos que necesitan tener un control interno de la calidad. Es por ello que se hace necesario revisar integralmente la calidad de cada uno de ellos.

En el centro CISED y CENIA no se realizan pruebas de liberación, trayendo como consecuencia que la versión entregada al LIPS, tenga n errores que no fueron previamente detectados, haciéndose el trabajo más engorroso para los probadores del LIPS y contando con un producto de poca calidad.

A pesar, de que si se realizan algunos tipos de pruebas, estas no se hacen con la seriedad requerida, ni su ejecución permite detectar la mayor cantidad de defectos posibles. Se entiende que la estrategia implementada en la facultad 1, no garantiza completamente el aseguramiento de la calidad del producto, y la mejora continua de su desarrollo, teniendo en cuenta que:

- La estrategia de aseguramiento de la calidad en los centros de la facultad no es la mejor, por lo que debe hacer énfasis en las pruebas que se le deben realizar al

Introducción

producto, para que este llegue al laboratorio industrial de pruebas (LIPS) con la menor cantidad de no conformidades posibles.

- En el centro CISED, el líder de proyecto no define diferentes tipos de pruebas para la liberación del producto.
- La documentación de los productos en ocasiones no está en correspondencia con el mismo.
- No se encuentra definido un procedimiento que rijan el proceso de liberación de los productos.
- No está implementado un modelo de calidad para la liberación del producto.

Esta investigación se enmarca en la fase intermedia, entre la terminación de la fase de implementación y el despliegue, donde se realizan las pruebas de liberación del producto. Esto es necesario porque no existe un laboratorio de pruebas, ni la facultad está en condiciones de implementarlo. Por lo tanto, el producto debe pasar primero por el laboratorio industrial de pruebas (LIPS), para que se le realicen las pruebas suficientes, y así detectar la mayor cantidad de no conformidades posibles quedando el mismo, libre de defectos. La propuesta permitirá obtener una forma de garantizar que los productos desarrollados tengan la menor cantidad de defectos posibles y que los centros cuenten con una guía para liberar sus productos.

Después de analizar la situación existente el siguiente **problema científico** es: ¿Cómo planificar, desarrollar y controlar las pruebas de liberación de los productos desarrollados en los centros de la facultad 1?

Por tanto, la presente investigación tiene como **objetivo general** proponer un procedimiento para las pruebas de liberación de los productos de software desarrollados en los centros de la facultad 1.

Se define que el **objeto de estudio** es el proceso de pruebas de calidad de los productos de software y el **campo de acción** puede identificarse como el proceso de pruebas de liberación de los productos desarrollados en los centros de la facultad 1.

Introducción

Para encaminar la investigación y con vistas a resolver el problema planteado se propone la siguiente **hipótesis**: Si se define un procedimiento para pruebas de liberación de los productos software, se garantizará el proceso de liberación de los productos.

Variable independiente: procedimiento para pruebas de liberación de los productos software

Variable dependiente: el proceso de liberación de los productos

Para satisfacer las necesidades planteadas es necesario establecer las **tareas de la investigación** que se exponen a continuación:

1. Análisis documental sobre los métodos y técnicas para la investigación.
2. Profundización del estudio de las normas enfocadas a la evaluación de la conformidad de los productos de software.
3. Diseño del procedimiento para el diseño, implementación y control de las pruebas de liberación de los productos de software de la facultad 1.
4. Validación de la propuesta del procedimiento de pruebas de liberación de los productos software desarrollados en los centros de la facultad 1.

El trabajo de diploma se encuentra estructurado de la siguiente manera:

Capítulo 1: "Fundamentación Teórica": como su título indica, fundamenta teóricamente el trabajo de diploma, en el que se hace un estudio de los tipos de pruebas, las pruebas de liberación, los elementos esenciales para la confección de un procedimiento para pruebas de liberación, cuyo resultado constituye la base del desarrollo de la presente investigación. Dentro de las principales técnicas y métodos a utilizar dentro del desarrollo de este capítulo están:

✓ **Métodos teóricos**

Histórico lógico: permite estudiar de forma analítica la trayectoria histórica real de los fenómenos, su evolución y desarrollo. Fue utilizado para estudiar todo lo referente a los métodos concernientes al tema.

Analítico-sintético: este método permite, partiendo de un análisis de toda la información recopilada, sintetizar y organizar de manera adecuada, para confeccionar una buena estructura

Introducción

buscando siempre la perfección. De esta manera se logrará un mayor entendimiento del procedimiento que se realizará.

Capítulo 2: “Diseño del procedimiento de pruebas para los productos de software desarrollados en los centros de la facultad 1”. En este apartado se utilizará como herramienta de apoyo en la definición de los problemas:

- ✓ Métodos empíricos:

Entrevistas: este método permitirá recopilar todo tipo de información con personas especialistas en el tema, para de esta forma enriquecer más la investigación.

Capítulo 3: “Validación del procedimiento propuesto” En este capítulo se valida la propuesta.

CAPÍTULO 1

Fundamentación teórica

En el presente capítulo se realiza el estudio del estado del arte, donde se brinda una visión general de los aspectos relacionados con la calidad de software, así como el aseguramiento de la calidad, la gestión de la calidad, entre otros. También se hace referencia sobre las diferentes normas y modelos que existen en función de la mejora continua, los niveles de pruebas que propone RUP que se utilizan para la revisión de un producto software, y otros tipos de pruebas de calidad.

Se investiga o evalúa todo el proceso de trabajo, los distintos laboratorios de calidad existentes en las diferentes facultades que existen en la Universidad de las Ciencias Informáticas (UCI), y experiencias que existen, tanto positivas como negativas. Así como un estudio de cómo se comporta fuera de la universidad este proceso, tanto en el ámbito nacional como internacional, tecnologías, métodos actuales, profundizando en las normas ISO establecidas para el control de la calidad del software, para así con esta investigación dar solución al problema planteado en la presente investigación.

1.1 Marco Conceptual

Prueba: es la actividad en la cual un sistema o uno de sus componentes, se ejecuta en circunstancias previamente especificadas donde los resultados se observan y registran y se realiza una evaluación de algún aspecto". Es el "proceso de ejecutar un programa con fin de encontrar errores". El nombre "prueba", además de la actividad de probar, se puede utilizar para designar "un conjunto de casos y procedimientos de prueba". (ISO 9000, 2005)

Pruebas de software: son los procesos que permiten verificar y revelar la calidad de un producto software. Se integran dentro de las diferentes fases del ciclo del software dentro de la Ingeniería de Software. Así se ejecuta un programa y mediante técnicas

experimentales se trata de descubrir qué errores tiene. Las mismas son el proceso de ejecución de un programa con la intención de descubrir un error. (ISO, 2008)

Pruebas basadas en experiencia: son aquellas pruebas que requieren de mayores competencias en el probador y están mayormente enfocadas a las experiencias obtenidas de pruebas anteriores.

- Adivinación de errores
- Ataques al sistema
- Exploratorias (Pressman, 1998)

Pruebas exploratorias: las pruebas exploratorias pueden ser realizadas en cualquier situación donde no sea obvio cual es la próxima prueba que se debe realizar. Puede tener varios objetivos como obtener retroalimentación rápida, de cierto producto o funcionalidad y aislar un defecto en particular, investigar el estado de un riesgo particular, o evaluar la necesidad de diseñar pruebas para determinada área. (Pressman, 1998).

Pruebas guiada por casos de prueba.

- Verifican las especificaciones funcionales y no consideran la estructura interna del programa.
- Se realiza sin el conocimiento interno del producto.
- No validan funciones ocultas (por ejemplo funciones implementadas pero no descritas en las especificaciones funcionales del diseño) por tanto los errores asociados a ellas no serán encontrados.

O sea los casos de prueba pretenden:

- Demostrar que las funciones del software son operativas.
- Que las entradas se aceptan de la forma adecuada y que se produce el resultado correcto.
- Así como la integración de la información externa (por ejemplo archivos de datos) se mantiene. (Pressman, 1998).

Pruebas de liberación: como política de la Universidad de las Ciencias Informáticas se aplican las pruebas de liberación. Su objetivo es revisar exhaustivamente un software y declararlo en estado óptimo para su entrega posterior al cliente y servir como antesala a las pruebas de aceptación. (Pressman, 1998)

Casos de prueba: el propósito de un caso de prueba es especificar una forma de probar el sistema, incluyendo las entradas con las que se ha de probar, los resultados esperados y las condiciones bajo las que ha de probarse. (Rational, 2003)

Modelos de calidad: es un conjunto de buenas prácticas para el ciclo de vida del software, enfocado en los procesos de gestión y desarrollo de proyectos. (Lewis, 2005)

Normas ISO: las normas son un modelo, un patrón, ejemplo o criterio a seguir. Una norma es una fórmula que tiene valor de regla y tiene por finalidad definir las características que debe poseer un objeto y los productos que han de tener una compatibilidad para ser usados a nivel internacional. La finalidad principal de las normas ISO es orientar, coordinar, simplificar y unificar los usos para conseguir menores costos y efectividad. Además, tiene valor indicativo y de guía. (Sans, 1998)

Evaluación de la conformidad: la evaluación de la conformidad es el procedimiento mediante el cual se determina el grado de cumplimiento de un proceso o producto, con las normas oficiales mexicanas, con las normas internacionales, o con otras especificaciones, prescripciones o características. La evaluación de la conformidad abarca, entre otros, los siguientes aspectos: muestreo, prueba, calibración, certificación y verificación. (ISO 9000, 2000)

1.2 Calidad del software

Actualmente el desarrollo de software se ha convertido en el principal aliado de las grandes empresas y organizaciones, pues el desarrollo y supervivencia de estas depende en gran medida de los sistemas informáticos. Además, el argumento de la calidad es exhibido por las empresas como un factor diferenciador, como clave de sus procesos de negocio y como eslogan de competitividad empresarial. Por lo que se hace hincapié en la necesidad de construir un software con una excelente calidad.

La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad, es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad. (Crosby, 2003).

Esto demuestra que un software debe tener relacionados enfoques de proceso, producto y efecto, a través de la aplicación de varios modelos y estándares como se

menciona anteriormente, además de garantizar que sea aceptado por clientes satisfechos, entonces, será un producto de software con calidad.

Con vistas a lograr la calidad del software, las empresas se enfocan en el alcanzar un Sistema de Gestión de la Calidad (SGC), y para ello se enfocan en: asegurar la calidad, gestionarla y controlarla.

1.3 Aseguramiento de la calidad

El aseguramiento de la calidad nace como una evolución natural del control de calidad, que resultaba limitado y poco eficaz para prevenir la aparición de defectos. Para ello, se hizo necesario crear sistemas de calidad que incorporasen la prevención como forma de vida y que, en todo caso, sirvieran para anticipar los errores antes de que estos se produjeran (otros, 2003)

El aseguramiento de la calidad: consiste en tener y seguir un conjunto de acciones planificadas y sistemáticas, implantadas dentro del sistema de calidad de la empresa. Estas acciones deben ser demostrables para proporcionar la confianza adecuada (tanto a la propia empresa como a los clientes) de que se cumplen los requisitos del sistema de la calidad. (Prof. Dr. Ing. Arturo Luis Romero, 2007)

Si se logra que se cumpla al pie de la letra el aseguramiento de la calidad en los centros pertenecientes a la facultad 1; se conseguirá entonces una mejor organización en el trabajo de estos. De esta forma, se logrará también una plena calidad de los productos desarrollados, así como satisfacer las necesidades del cliente.

1.4 Gestión de la calidad

La gestión de calidad tiene que ver con la organización interna que ejerce la determinación de los procesos productivos y de las características y cualidades de los productos, es decir, es la gerencia o el manejo de los procesos productivos enfocada al mejoramiento continuo (Crosby, 2003).

La gestión de la calidad está orientada hacia los deseos del cliente, tiene a los clientes en primer plano, espera resultados positivos y a largo plazo, cuida la imagen de la empresa, involucra a todos los niveles jerárquicos, la interfaz, el proceso, el potencial, una cultura, una filosofía, es un proceso que destaca la responsabilidad de todos, y de cada uno de los individuos, aspira a la mejora continua.

En fin se puede aportar que el uso de un sistema de gestión de calidad dentro de una compañía es positivo para todos: empresarios, trabajadores y clientes, y como tal su misión es la de: establecer la política y los objetivos para obtener resultados en base a una organización o compañía con el objetivo final de alcanzar un grado de calidad. (Lewis, 2005). Para una correcta gestión de la calidad, se debe tener en cuenta los siguientes conceptos: planificar, organizar, dirigir y controlar las actividades para conseguir los objetivos fijados.

La gestión de la calidad se basa en los 8 principios de calidad. (UNITEC., 2003)

1.- Enfoque al cliente: las organizaciones dependen de sus clientes, por lo tanto, deben comprender sus necesidades actuales y futuras, satisfacer sus requisitos y esforzarse en exceder sus expectativas.

2.- Liderazgo: los líderes establecen la unidad de propósito y la orientación de la organización. Deben crear y mantener un ambiente interno, en el cual el personal pueda llegar a involucrarse en el logro de los objetivos de la organización.

3.- Participación del personal: el personal, a todos los niveles, es la esencia de la organización, y su total compromiso posibilita que sus habilidades sean usadas para el beneficio de la organización.

4.- Enfoque basado en procesos: un resultado deseado se alcanza más eficientemente cuando las actividades y los recursos relacionados se gestionan como un proceso. Ver siguiente capítulo para conocer más sobre los procesos.

5.- Enfoque de sistema para la gestión: identificar, entender y gestionar los procesos interrelacionados como un sistema, contribuye a la eficacia y eficiencia de la organización en el logro de sus objetivos.

6.- Mejora continua: la mejora continua del desempeño global de la organización, debe de ser un objetivo permanente de esta.

7.- Enfoque basado en hechos para la toma de decisiones: las decisiones eficaces se basan en el análisis de los datos y en la información previa.

8.- Relaciones mutuamente beneficiosas con el proveedor: una organización y sus proveedores son interdependientes, y una relación mutuamente beneficiosa aumenta la capacidad de ambos para crear valor.

1.5 Control de la calidad

El control de calidad se realiza o participa en la caracterización de los nuevos productos en sus diferentes fases de desarrollo y en el establecimiento de las especificaciones de calidad de los mismos. Desarrolla, ejecuta o coordina la ejecución de los métodos de ensayo para determinar las características de calidad de las materias primas, materiales, productos intermedios y productos finales. (19011:2002)

1.6 Sistema de Calidad

Un sistema de calidad se centra en garantizar que lo que ofrece una organización cumple con las especificaciones establecidas previamente por la empresa y el cliente, asegurando una calidad continua a lo largo del tiempo. (Manuel García, 2003)

1.7 Ciclo Deming

Uno de los principios básicos de la calidad es la prevención y la mejora continua. Esto significa que la calidad es un proyecto interminable, cuyo objetivo es detectar disfunciones tan rápidas como sea posible después de que ocurran. Así, la calidad puede representarse en un ciclo de acciones correctivas y preventivas llamado “ciclo de Deming”. (herramientasparapymes, 2005)

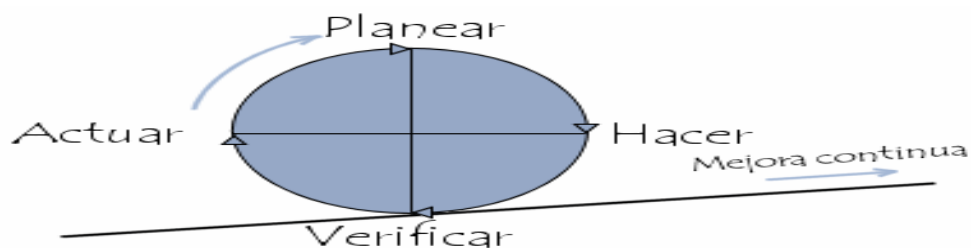


Figura 1.1: Representación del Ciclo Deming

Este ciclo, representado en el ciclo de Deming, se llama modelo PDCA. PDCA se refiere a las iniciales del inglés de los siguientes cuatro pasos:

Planear (*plan*): definir los objetivos a alcanzar y planificar cómo implementar las acciones.

Hacer (*do*): implementar las acciones correctivas

Controlar: (*Check*): verificar que se logre el conjunto de objetivos.

Actuar (*act*): según los resultados obtenidos en el paso anterior, tomar medidas preventivas.

Este ciclo es utilizado entre otras cosas para la mejora continua de la calidad dentro de una empresa. El método consiste en aplicar 4 pasos definidos para asegurar alcanzar el objetivo definido. Estos 4 pasos aseguran par el proyecto:

- La organización lógica del trabajo.
- La correcta realización de las tareas necesarias y planificadas.
- La comprobación de los logros obtenidos.
- La posibilidad de aprovechar y extender aprendizajes y experiencias adquiridas a otros casos. (herramientasparapymes, 2005)

1.8 Mejora continua

La mejora continua es una herramienta de incremento de la productividad. Algunas de las herramientas utilizadas incluyen las acciones correctivas, preventivas y el análisis de la satisfacción en los miembros o clientes. Se trata de la forma más efectiva de mejora de la calidad y la eficiencia en las organizaciones. (Manuel García, 2003). Es muy recomendable que la mejora continua sea vista como una actividad sostenible en el tiempo y regular y no como un arreglo rápido frente a un problema puntual.

Se ha analizado la importancia de cumplir con el SGC y garantizar la mejora continua de los procesos y los productos de la empresa. En este caso la UCI como productora de software debe tener en cuenta estos criterios para garantizar su mejora continua. El análisis del SGC asegura que la propuesta esté enfocada en garantizar la calidad de los procesos de la Universidad y de sus productos. A continuación se hace un análisis de las metodologías utilizadas en los centros de la facultad 1.

1.9 Pruebas de software

Las pruebas de software constituyen uno de los procesos fundamentales dentro del control de calidad del software. Consiste en la ejecución de un programa bajo ciertos datos de entrada (casos de prueba), para posteriormente comparar las salidas obtenidas con las deseadas (función para la cual fue diseñado). (Pressman, 1998).

Se puede decir que las pruebas de software son el proceso de establecer la existencia de errores en un programa o sistema; que nunca demuestra que un programa es

correcto, siempre es posible que existan errores aún después de la prueba más completa. Las pruebas de software sólo pueden demostrar la presencia de errores, no su ausencia.

Existen varios niveles de prueba definidos y estos niveles a su vez especifican qué tipos y métodos de pruebas se deben utilizar en cada uno de ellos y cuáles son sus objetivos. A continuación se hace referencia a los niveles de pruebas: (Rational, 2003)

- Prueba de desarrollador
- Prueba independiente
- Prueba de unidad
- Prueba de integración
- Prueba de sistema
- Prueba de aceptación

Prueba de Desarrollador. Es la prueba diseñada e implementada por el equipo de desarrollo. Tradicionalmente estas pruebas han sido consideradas solo para la prueba de unidad, aunque en la actualidad en algunos casos pueden ejecutar pruebas de integración. Se recomienda que estas pruebas cubran más que las pruebas de unidad.

Prueba independiente: es la prueba que es diseñada e implementada por alguien independiente del grupo de desarrolladores. El objetivo de estas pruebas es proporcionar una perspectiva deferente y en un ambiente más rico que los desarrolladores. Una vista de la prueba independiente es la prueba independiente de los interesados, que son pruebas basadas en las necesidades y preocupaciones de los interesados.

Tipo de pruebas independiente

- Prueba de liberación
 1. Prueba el software funcionando como un todo.
 2. Trata de probar que los objetivos para los que fue construida la aplicación no se cumplen en su totalidad y que por tanto el software no está en condiciones de ser presentado al cliente y hay que hacer modificaciones a la aplicación.
 3. Se realiza cuando el software funciona como un todo y está prácticamente listo para ser presentado al cliente.

Prueba de unidad: es la prueba enfocada a los elementos probables más pequeños del software. Es aplicable a componentes representados en el modelo de

implementación para verificar que los flujos de control y de datos están cubiertos, y que ellos funcionen como se espera. La prueba de unidad siempre está orientada a caja blanca.

Antes de iniciar cualquier otra prueba es preciso probar el flujo de datos de la interfaz del componente. Si los datos no entran correctamente, todas las demás pruebas no tienen sentido. El diseño de casos de prueba de una unidad comienza una vez que se ha desarrollado, revisado y verificado en su sintaxis el código a nivel fuente.

1. Enfocadas al código fuente de los componentes.
2. Para verificar todos los flujos de control.
3. Primero pasa por la revisión del programador.

La práctica que se utiliza para desarrollar las pruebas de unidad es: Desarrollo Dirigido por Pruebas (TDD): es una técnica de programación que plantea escribir primero los casos de prueba y luego implementar el necesario para ejecutarla. Los mayores beneficios de TDD son:

Antes de escribir cualquier fragmento de código, se debe escribir las pruebas automatizadas para comprobar la funcionalidad de ese futuro código. Como el código no existe todavía, inicialmente la prueba falla.

Una vez comenzadas las pruebas, se debe eliminar el código duplicado.

Prueba de integración: prueba los componentes combinados para ejecutar un Caso de Uso (CU) y para verificar y descubrir errores en las especificaciones de las interfaces de las clases.

Es ejecutada para asegurar que los componentes en el modelo de implementación operen correctamente cuando son combinados para ejecutar un caso de uso. Se prueba un paquete o un conjunto de paquetes del modelo de implementación. Estas pruebas descubren errores o incompletitud en las especificaciones de las interfaces de los paquetes. Esta prueba debe ser responsabilidad de desarrolladores y de independientes, sin solaparse las pruebas.

Es el proceso de combinar y probar múltiples componentes juntos. El objetivo es tomar los componentes probados en unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño.

Se llama integración incremental cuando el programa se construye y se prueba en pequeños segmentos en los que los errores son más fáciles de aislar y corregir, es más probable que se pueda probar completamente las interfaces y se puede aplicar un enfoque de prueba sistemática.

Hay dos estrategias de integración incremental:

Prueba de Regresión: pruebas orientadas a descubrir la causa de nuevos errores.

Las pruebas de sistema están divididas en dos grandes grupos:

- ✓ Funcionales: Enfocadas a los requisitos funcionales del software, a su interacción con el cliente de la forma que ha sido pactada.
- ✓ No funcionales: Enfocadas a los requisitos no funcionales del proyecto. desglosadas en un número de pruebas definidas como:
 1. Prueba de seguridad: Los sistemas y políticas de seguridad son analizados exhaustivamente con el fin de encontrar fallos de seguridad, tanto en el diseño, como en la implementación de la aplicación.
 2. Prueba de disponibilidad y red: Verifica de manera continua que el entorno está funcionando como espera y su disponibilidad está dentro de los límites que han sido previamente establecidos.
 3. Prueba de rendimiento o carga: Se ejecutan tanto para determinar como responde un sistema ante una cierta carga, como para validar otros atributos relacionados con la calidad, como pueden ser la escalabilidad o el uso de recursos, entre otros.
 4. Prueba de compatibilidad: Ayudan a determinar si el producto funcionará correctamente con otro hardware y software en el entorno pretendido.
 5. Prueba de resistencia o estrés: Consiste en poner N usuarios virtuales accediendo concurrentemente durante varias horas y verificar que la aplicación responde a las peticiones en un tiempo menor a T.
 6. Prueba de usabilidad: Consiste en asegurar que se identifiquen y corrijan a tiempo los fallos en la interfaz gracias al seguimiento de los comentarios de los usuarios finales.
 7. Prueba de fiabilidad: Verifica la probabilidad de que un programa realice su objetivo satisfactoriamente (sin fallos) en un determinado período de tiempo y en un entorno concreto (denominado perfil operacional).

Prueba de Sistema: es la actividad de prueba dirigida a verificar el programa final, después que todos los componentes de software y hardware han sido integrados.

En un ciclo iterativo estas pruebas ocurren más temprano, tan pronto como subconjuntos bien formados de comportamiento de caso de uso son implementados.

1. Prueba el software funcionando como un todo.
2. Aceptable para cuando el software se encuentra en la Fase de Construcción.
3. Trata de probar que los objetivos para los que fue construida la aplicación no se cumplen en su totalidad y que por tanto hay que cambiar cosas en la aplicación.
4. Se usa como base los objetivos originales.
5. No existe un método en específico, sino que se dan lineamientos, a la hora de preparar los casos de prueba.
6. Se finaliza cuando se cumplieron los meses o las semanas programadas y se han hallado n errores.

Tipo de Pruebas del Sistema

- Prueba de Recuperación: es una prueba del sistema que fuerza el fallo del software de muchas formas y verifica que la recuperación se lleva a cabo apropiadamente.
- Prueba de Seguridad: intenta verificar que los mecanismos de protección incorporados en el sistema lo protegerán, de hecho, de accesos impropios.
- Prueba de Resistencia: están diseñadas para enfrentar a los programas con situaciones anormales.
- Prueba de Rendimiento: Está diseñada para probar el rendimiento del software en tiempo de ejecución dentro del contexto de un sistema integrado.

Prueba de aceptación: Prueba de aceptación del usuario es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido. Prueba el software funcionando como un todo, se realiza dándole un uso real a la aplicación y es realizada por parte de los clientes, los errores que se encuentren en la misma son reportados como defectos

Un comentario sobre la secuencia de los niveles de prueba.

Las pruebas de unidad son implementadas en la iteración más temprana como el primer nivel de prueba. Pero en un proceso iterativo ejecutar todas las pruebas de unidad antes de pasar a niveles siguientes de prueba como regla es inapropiado. Una mejor estrategia es identificar las pruebas de unidad, integración y sistema que

ofrecen mayor potencial para encontrar errores y entonces implementarlas y ejecutarlas.

Otros Tipos de pruebas

Prueba de funcionalidad: El objetivo de estas pruebas es verificar la aceptación, procesamiento y recuperación de datos y la adecuada implementación de las reglas de negocio. Estos tipos de pruebas, están basadas en técnicas de caja negra, es decir, verificar la aplicación interaccionando a través de las interfaces de usuario y analizando los resultados.

Prueba de volumen: validar que el sistema funcione con el máximo número de usuarios conectados y acceso en gran escala a la base de datos.

Prueba de carga: este es el tipo más sencillo de pruebas de rendimiento. Una prueba de carga se realiza generalmente para observar el comportamiento de una aplicación bajo una cantidad de peticiones esperada. Esta carga puede ser el número esperado de usuarios concurrentes utilizando la aplicación y que realizan un número específico de transacciones durante el tiempo que dura la carga. Esta prueba puede mostrar los tiempos de respuesta de todas las transacciones importantes de la aplicación. Si la base de datos, el servidor de aplicaciones, etc, también se monitorizan, entonces esta prueba puede mostrar el cuello de botella en la aplicación.

Prueba de performance: Su objetivo es validar los tiempos de respuesta bajo 2 condiciones:

- ✓ Volumen Normal -Peores Condiciones.

Prueba de configuración: Su objetivo es probar un sistema de software con distintas combinaciones de hardware y software. La manera tradicional de realizar pruebas de configuración de software consiste en configurar un laboratorio con varios equipos físicos que ejecutan diferentes combinaciones de sistemas operativos, exploradores Web y otro software. Es decir, estas pruebas se enfocan en evaluar las diferentes variaciones de una aplicación integrada, contra sus requerimientos de configuración.

Una de sus desventajas es que esto puede resultar caro y llevar mucho tiempo.

Prueba de instalación: Su objetivo es asegurar el funcionamiento correcto de opciones y funcionalidades de la instalación. También se realiza para asegurar que todos los componentes necesarios sean realmente instalados.

Se exponen todos los tipos de pruebas existentes, sin embargo, el trabajo de diploma se enfoca específicamente en las pruebas de liberación. A continuación se presenta más detalladamente las características de este tipo de prueba.

Prueba de stress: enfocada a evaluar cómo el sistema responde bajo condiciones anormales. (Extrema sobrecarga, insuficiente memoria, servicios y hardware no disponible, recursos compartidos no disponible)

Prueba de benchmark (comprobador de rendimiento): es un tipo de prueba que compara el rendimiento de un elemento nuevo o desconocido a uno de carga de trabajo de referencia conocido.

Prueba de usabilidad: prueba enfocada a factores humanos, estéticos, consistencia en la interfaz de usuario, ayuda sensitiva al contexto y en línea, asistente documentación de usuarios y materiales de entrenamiento.

Prueba de seguridad: Asegurar que los datos o el sistema solamente es accedido por los actores deseados.

1.9.1 Pruebas de liberación

El objetivo de estas pruebas es demostrar al desarrollador y al cliente que el software satisface los requisitos. Descubrir defectos en el software con respecto a su comportamiento, el cual puede ser incorrecto, no deseable o no cumple su especificación.

Mundialmente, productores de software tienen sus propias fórmulas, tanto técnicas como comerciales, para decidir cuándo liberar sus productos y llevarlo al producto real. Pero siempre se teme a un riesgo, a una complicación del negocio. (guiadelacalidad, 2000)

Se ha llevado a cabo un estudio para evaluar el riesgo de liberar el software defectuoso que no puede ser probado exhaustivamente y sin necesidad de pruebas de defectos de software libre. Sólo hay tantas pruebas beta que se puede hacer, después de todo.

Los procedimientos de prueba utilizados son imperfectos y, a continuación, es relativamente fácil para una empresa liberar un producto que piensan que es casi perfecto, sólo para descubrir que los usuarios reales que trabajan con el software en un entorno real puedan encontrar aún disímiles errores.

Prueba de liberación es un término usado en la UCI, y se definen como un conjunto de pruebas que se realizan a todo software producido en la misma, para avalarlo como apto para entregarse al cliente.

El sistema no es el único artefacto a probar, su documentación también forma parte del campo de acción de las pruebas, principalmente aquellos documentos que son entregados junto al producto para guiar y ayudar al cliente a interactuar con el programa.

En la universidad existe un laboratorio industrial de pruebas (LIPS), donde se realizan las pruebas de liberación de los productos de exportación que se desarrollan en los diferentes centros de la Universidad, como son las aplicaciones Desktop y Web.

El método que se propone a utilizar es el de caja negra, utilizando la técnica de particiones equivalentes propuesto por el Proceso Unificado de Desarrollo (*RUP* por sus siglas en inglés), al que se le han añadido algunas consideraciones, teniendo en cuenta la experiencia adquirida por los autores de este trabajo.

1.9.2 Método de prueba

Prueba de caja negra (Rational, 2003)

Prueba de caja negra: se refiere a las pruebas que se llevan a cabo sobre la interfaz del software, por lo que los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene. Esta prueba examina algunos aspectos del modelo fundamentalmente del sistema sin tener mucho en cuenta la estructura interna del software.

Dentro de las pruebas de liberación para la presente investigación se utiliza específicamente el método de caja negra, el cual se centra principalmente en los requisitos funcionales del software. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos.

Las pruebas de caja negra:

1. Verifican las especificaciones funcionales y no consideran la estructura interna del programa.

2. Es hecha sin el conocimiento interno del producto.
3. No validan funciones ocultas (por ejemplo funciones implementadas pero no descritas en las especificaciones funcionales del diseño) por tanto los errores asociados a ellas no serán encontrados.

En otras palabras, la prueba de la caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, los casos de prueba pretenden:

- Demostrar las funciones del software son operativas.
- Que las entradas se aceptan de la forma adecuada y que se produce el resultado correcto.
- Así como la integrada de la información externa (por ejemplo archivos de datos) se mantiene.

Muchos autores consideran que estas pruebas permiten encontrar como: (Pressman, 1998)

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

Para preparar los casos de pruebas hacen falta un número de datos que ayuden a la ejecución de los estos casos y que permitan que el sistema se ejecute en todas sus variantes, pueden ser datos válidos o inválidos para el programa según si lo que se desea es hallar un error o probar una funcionalidad. Los datos se escogen atendiendo a las especificaciones del problema, sin importar los detalles internos del programa, a fin de verificar que el programa corra bien.

Para desarrollar la prueba de caja negra existen varias técnicas, entre ellas están:

- Técnica de la partición de equivalencia: esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software. Esta será la técnica a utilizar para el desarrollo de la investigación.
- Técnica del análisis de valores límite: esta técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.

- Técnica de grafos de causa-efecto: es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

1.10 Modelos de calidad

1.10.1 Importancia de la evaluación y medición de la calidad del software

A medida que el desarrollo de los productos de software ha ido evolucionando, también lo han hecho los procesos evaluativos que lo acompañan.

A pesar del gran número de artículos de investigación y normas existentes sobre el tema de validación de calidad del producto de software, en la actualidad existen muy pocas industrias de software que utilicen procesos de evaluación y análisis para este efecto. Actualmente, la gran mayoría de estudios están enfocados a las actividades de administración de los proyectos de desarrollo de software.

En diversos entornos industriales y académicos, la calidad del software ha sido evaluada (validada) mediante distintos estudios analíticos. De dichos entornos, la evaluación de la calidad del software ha evolucionado hacia modelos formales estadísticos que se basan en métricas como fundamento para el aseguramiento, control y evaluación de la calidad de un producto o proceso de software. (Tejeda, 2007)

Cuando se evalúa, el objetivo que se persigue es realizar revisiones durante el proceso de desarrollo para detectar los errores que afecten a la operación del producto. Es claro que un buen análisis de calidad es una de las bases más importantes para la toma de decisiones dentro de una organización que desarrolla software de calidad. Por tal razón, la evaluación de un producto de software permitirá mejorar el control de la calidad de un producto de software (Leticia Dávila Nicanor)

La medición cuenta con una larga tradición y constituye una disciplina fundamental en cualquier ingeniería, y la Ingeniería del Software no debe ser una excepción, si bien hay que tener siempre presente las peculiaridades que diferencian al software de otros productos.

La medición del software es una disciplina joven, y ello ha influido notablemente en que la Ingeniería del Software no haya alcanzado aún el grado de madurez que tienen otras ingenierías.

En la actualidad pocos dudan de la importancia de la medición para conseguir incrementar la calidad y la productividad en el desarrollo y mantenimiento del software. La necesidad y motivación por medir se ha incrementado notablemente con la preocupación de las organizaciones por alcanzar mayores niveles de madurez y las consiguientes certificaciones basadas en modelos y normas como ISO 9000, ISO 15504 o CMMI.

1.10.2 Normas ISO

La Organización Internacional de Normalización (ISO), con sede en Ginebra, Suiza, nació en 1947. Desde entonces, adoptó como nombre oficial el vocablo ISO que es símbolo de igualdad y estandarización a escala internacional. (Cruz, 2008)

En 1987 se establecen las normas ISO 9000 para el aseguramiento de la calidad y es concebida como una metodología de procesos basada en los requisitos a cumplir para garantizar la calidad del producto y de los servicios.

En 1904 se hizo una primera revisión de las normas ISO, las cuáles fueron actualizadas en el 2000, y en el 2008 salió la cuarta edición de las Normas ISO 9001.

Las normas son un modelo, un patrón, ejemplo o criterio a seguir. Una norma es una fórmula que tiene valor de regla y tiene por finalidad definir las características que debe poseer un objeto y los productos que han de tener una compatibilidad para ser usados a nivel internacional. La finalidad principal de las normas ISO es orientar, coordinar, simplificar y unificar los usos para conseguir menores costos y efectividad, además tiene valor indicativo y de guía. (19011:2002)

Entre estas normas tenemos la norma ISO 9000 que no es más que un conjunto de normas de calidad establecidas por la ISO que se pueden aplicar en cualquier tipo de organización que buscan ventajas por medio de la implementación de un Sistema de Gestión de la Calidad (SGC); las organizaciones que buscan la confianza de sus proveedores en que sus requisitos para los productos serán satisfechos; los usuarios de los productos.

Todas estas normas juntas forman un conjunto coherente de normas de SGC que facilitan la mutua comprensión en el comercio nacional e internacional la familia de normas ISO 9000 citadas a continuación. Se han elaborado para asistir a las

organizaciones, de todo tipo y tamaño, en la implementación y la operación de sistemas de gestión de la calidad eficaz.

La familia de ISO 9000 está compuesta por: (ISO 9000, 2005)

- La Norma **ISO 9000 del 2000** que describe los fundamentos de los sistemas de gestión de la calidad y especifica la terminología para los sistemas de gestión de la calidad. Esta norma utiliza además los 8 principios de gestión de la calidad utilizados por la alta dirección con el fin de conducir a la organización hacia una mejora en el desempeño.
- La Norma **ISO 9000 del 2005** esta tercera edición anula y reemplaza a la segunda edición (ISO 9000:2000). Incluye los cambios aceptados en el borrador de modificación ISO/DAM 9000:2004. Esta norma Internacional describe los fundamentos de los sistemas de gestión de la calidad, los cuales constituyen el objeto de la familia de Normas ISO 9000, y define los términos relacionados con los mismos.
- La Norma **ISO 9001 del 2008** especifica los requisitos para un SGC que pueden utilizarse para su aplicación interna por las organizaciones, para certificación o con fines contractuales. Se centra en la eficacia del sistema de gestión de la calidad para satisfacer los requisitos del cliente. Esta cuarta edición anula y sustituye a la tercera edición (ISO 9001:2000), que ha sido modificada para clarificar puntos en el texto y aumentar la compatibilidad con la Norma ISO 14001:2004.
- **ISO 9002:** Sistema donde las características del producto o servicio son definidas por el cliente. Modelo para el aseguramiento de la calidad en la producción, instalación y servicio postventa. (ISO, 2008)
- **ISO 9003:** Modelo para el aseguramiento de la calidad en la inspección y los ensayos finales. Cubre las obligaciones de aseguramiento de calidad en las áreas de control final y pruebas. Es de limitada aplicación por lo que existen planes para su eliminación.
- La Norma **ISO 9004** proporciona directrices sobre un rango más amplio de objetivos de un sistema de gestión de la calidad que la Norma ISO 9001, especialmente para la mejora continua del desempeño y eficiencia global de la organización. La Norma ISO 9004 se recomienda como una guía para aquellas

organizaciones cuya alta dirección desee ir más allá de los requisitos de la Norma ISO 9001, en busca de la mejora continua del desempeño. Sin embargo, no está pensada para su utilización con fines contractuales o de certificación.

Estas normas están enfocadas en alcanzar en la empresa un SGC, para controlar y asegurar la calidad, pero no responden a evaluar los productos a partir de una serie de características con el objetivo de tener un amplio y fuerte criterio para liberar los productos.

Actualmente se está creando por la ISO normas enfocadas más al producto y a su medición, pero aún no han sido adoptadas por el estado cubano.

En Cuba, actualmente existen normas para la evaluación del producto como la ISO/IEC 14598, la ISO/IEC 25000:2005, sin embargo en la presente investigación se hace énfasis NC ISO/IEC 9126. Debido a que este modelo de calidad es para la especificación y evaluación del producto software. La misma está dividida en 4 partes:

- NC-ISO/IEC 9126-2: Calidad externa
- NC-ISO/IEC 9126-3: Calidad interna
- NC-ISO/IEC 9126-4: Calidad en el uso

A partir de la necesidad de evaluar el producto como parte de su liberación, la investigación se basa específicamente en la norma cubana NC-ISO/IEC 9126 del 2005 en su parte 1, denominada modelo de la calidad.

1.10.2.1 Norma cubana NC ISO/IEC 9126-1:2005 (9126., 2005)

Enfocada en el producto de software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas cuando el software se usa bajo las condiciones, tales como: algunos aspectos de funcionalidad, confiabilidad y eficiencia que también influirán la usabilidad, pero para los propósitos de la NC ISO/IEC 9126-1:2005, no son clasificados dentro de la usabilidad.

Esta parte de la NC ISO/IEC 9126 permite especificar y evaluar la calidad del producto de software desde las perspectivas de aquellos asociados con la adquisición, regulación, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoría del software. Por ejemplo, puede ser utilizada por los programadores, los clientes, el personal de aseguramiento de la calidad y los evaluadores independientes, particularmente los responsables de especificar y evaluar la calidad de los productos de software.

El modelo de calidad definido en esta parte puede usarse para:

- validar la integridad de la definición de los requisitos;
- identificar los requisitos del software;
- identificar los objetivos del diseño del software;
- identificar los objetivos de ensayo del software;
- identificar los criterios de aseguramiento de la calidad;
- identificar los criterios de aceptación para un producto de software terminado.

Se hace necesario destacar que esta parte de la NC ISO/IEC 9126 se puede utilizar junto con la ISO/IEC 9001 (que se relaciona con los procesos de gestión de la calidad) para garantizar:

- un apoyo para establecer metas de calidad,
- un apoyo para la revisión, verificación y validación del diseño.

Esta parte de la NC-ISO/IEC 9126 describe un modelo en dos partes para calidad de los productos de software: a) calidad interna y externa y b) calidad durante el uso. La primera parte del modelo especifica seis características para la calidad interna y externa, que son además divididas en subcaracterísticas que se manifiestan externamente cuando el software se usa como una parte del sistema computarizado, y son un resultado de los atributos internos del software.



Figura 1.2: Características y subcaracterísticas de la calidad interna y externa

En la práctica no es imposible medir todas las subcaracterísticas internas o externas para todas las partes de un producto de software de gran tamaño. Igualmente no es práctico medir la calidad en el uso para todos los posibles escenarios de tareas de usuario. Los recursos para la evaluación se asignarán entre los tipos diferentes de medida según los objetivos comerciales y la naturaleza del producto así como el proceso de diseño. A continuación se describen cada una de estas características.

1. Funcionalidad: un sistema de las cualidades que refieren la existencia de un sistema de funciones y de sus características especificadas. Las funciones son las que satisfacen necesidades indicadas o implicadas.

- Idoneidad
- Precisión
- Interpolaridad
- Funcionalidad
- Conformidad

2. Confiabilidad: un sistema de las cualidades que refieren la capacidad del software para mantener su nivel del funcionamiento bajo condiciones indicadas por un período del tiempo indicado.

- Madurez
- Tolerancia al defecto
- Recuperabilidad
- Fiabilidad
- Conformidad

3. Usabilidad: un sistema de las cualidades que refieren el esfuerzo que necesitó para el uso, y en la carga individual de tal uso, por un sistema indicado o implicado de usuarios.

- Instructibilidad
- Comprensibilidad
- Operabilidad
- Atractivo
- Utilizabilidad

- Conformidad

4. Eficacia: un sistema de las cualidades que refieren la relación entre el nivel del funcionamiento del software y la cantidad de recursos usados, bajo condiciones indicadas.

- Comportamiento de tiempo
- Utilización de los recursos
- Eficacia
- Conformidad

5. Mantenibilidad: un sistema de las cualidades que refieren el esfuerzo que se necesitó para hacer modificaciones especificadas.

- Estabilidad
- Analizabilidad
- Cambiabilidad
- Ensayabilidad
- Mantenibilidad
- Conformidad

6. Portabilidad: un sistema de las cualidades que refieren la capacidad del software de ser transferido de un ambiente a otro.

- Instalabilidad
- Reemplazabilidad
- Adaptabilidad
- Coexistencia
- Portabilidad
- Conformidad

La segunda parte del modelo especifica cuatro características de calidad durante el uso, pero no elabora dicho modelo a un nivel inferior al de las características. La calidad durante el uso es el efecto combinado para el usuario de las seis características de calidad del producto de software.

Las características definidas en su parte 1, son aplicables a todo tipo de software, incluidos los programas de computación y los datos contenidos en el *firmware*¹

(programación en firme) (programación en firme). Las características y sub-características proveen una consistente terminología sobre la calidad del software y además proveen un marco para especificar los requisitos de calidad para el software permitiendo el intercambio entre las diversas capacidades del producto de software.

1.11 Experiencias en la evaluación de los productos en los centros

La NC ISO/IEC 9126-1:2005 define sus características para cualquier producto, por lo que estas son aplicables a las aplicaciones Web desarrolladas por los centros. Como se define en las especificaciones de la norma, evaluar cada una de estas características permite tener una medida de la conformidad del producto.

En los proyectos del centro Identificación y Seguridad Digital (CISED) y el centro de Informatización Universitaria (CENIA), se realizan pruebas al producto, sin embargo, las pruebas no se enfocan en su totalidad en evaluar el producto, por lo que en su generalidad no se aplica un modelo de calidad que garantice el éxito para obtener un producto de alta calidad.

A pesar de que el LIPS utiliza esta norma y ha enfocado algunas pruebas a la evaluación, los centros de la facultad 1, al no contar con una estrategia para liberar sus productos, tampoco guían sus esfuerzos hacia este sentido. Por lo que no tienen mucha experiencia en la aplicación de este modelo y en sacar el mejor provecho del mismo.

Para la investigación se realizaron diferentes tipos de análisis para determinar las causas de cómo se realiza el proceso de pruebas en los centros, con vistas a enfocar la propuesta, basada en la NC ISO/IEC 9126-1:2005.

1.12 Análisis de causas y efectos en la preparación de un producto en la facultad 1 aplicando el método de Ishikawa.

El diagrama causa-efecto conocido también como diagrama de espina de pescado es utilizado para facilitar el análisis de problemas y sus soluciones en esferas, como lo son: calidad de los procesos, los productos y servicios.

Este diagrama es utilizado además para ayudar a pensar todas las causas, ya sean las más simples o más complejas que posea un problema. Permite comprender el problema, identificar posibles soluciones del mismo, tomar decisiones y poder organizar los futuros planes a realizar.

A continuación se muestra el diagrama causa-efecto que muestra las causas que provocan que haya poca calidad al realizar las pruebas de liberación en la facultad 1. Estos problemas se detectaron a partir de entrevistas realizadas a 25 proyectos pertenecientes a los centros de la facultad 1. Estas fueron dirigidas específicamente a los líderes de proyecto, asesores de calidad, programadores y analistas, por lo que el diagrama muestra una visión del estado de las pruebas en la facultad. Teniendo en cuenta que el centro de Informatización Universitaria (CENIA) tiene un total de 33 proyectos y el centro de Identificación y Seguridad Digital (CISED) tiene alrededor de 15 proyectos el porcentaje que representa la cantidad de proyectos entrevistados del total es 52 %.

Diagrama Causa-Efecto.

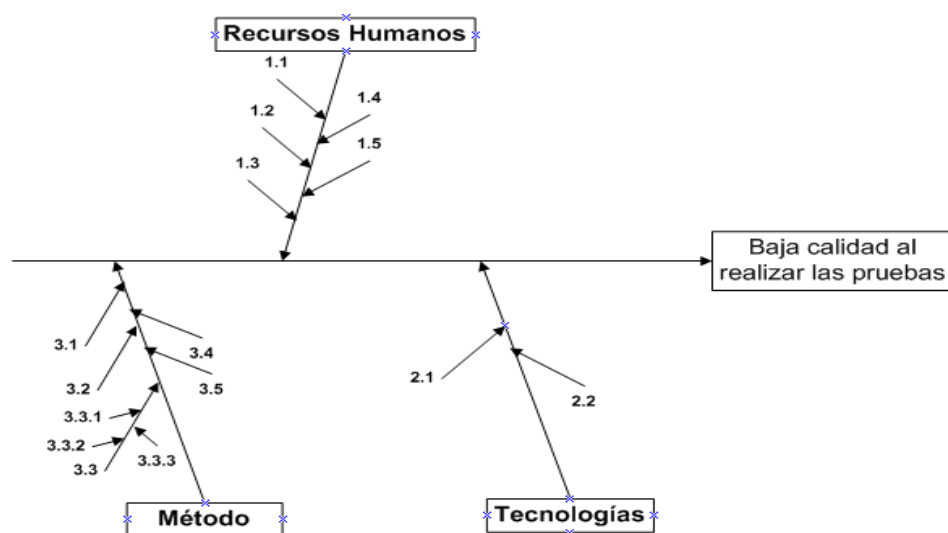


Figura 1.3: Diagrama de Causa-Efecto

Leyenda:

Efecto: Baja calidad en la realización de las pruebas

1. Recursos Humanos

- 1.1. Poca comunicación entre los probadores y el equipo de desarrollo del proyecto.
- 1.2. Poco conocimiento del equipo probador del proceso de pruebas que se debe realizar.

- 1.3. Omisión de pasos y tipos de pruebas por inexperiencias o facilísimo.
- 1.4. Poca identificación del equipo probador con el proceso de pruebas que realiza.
- 1.5. No se cuenta con la cantidad de personas necesarias para la realización de las pruebas en la facultad.

2. Tecnologías

- 2.1. La facultad no cuenta con un laboratorio para realizar las pruebas.
- 2.2. Las computadoras en ocasiones asignadas no tienen el rendimiento necesario para realizar algunos tipos de pruebas.

3. Método

- 3.1. No existe un método o procedimiento específico para realizar las pruebas.
- 3.2. La facultad no cuenta con un equipo de trabajo capacitado para esta tarea.
- 3.3. Las pruebas que se ejecutan no son lo suficientemente óptimas
 - 3.3.1. No se realizan las pruebas según los criterios de sus funcionalidades.
 - 3.3.2. Los casos de pruebas no son diseñados correctamente.
 - 3.3.3. No se le da seguimiento a las no conformidades que surgen de las iteraciones de pruebas.
- 3.4. No se realizan las pruebas pertinentes.
- 3.5. Inexistencia de un registro de experiencias anteriores como guía para la realización de las pruebas de liberación en la facultad.

1.13 Conclusiones Parciales

A partir de la necesidad que tiene la facultad 1 de más que contar con un laboratorio, tener el personal capacitado para llevar a cabo las pruebas y garantizar, que los productos que desarrollan cuenten con la calidad necesaria y que la versión a desplegar tenga la menor cantidad de defectos posibles, se define realizarle diferentes tipos de pruebas al software, con vistas a evaluarlo, identificar no conformidades y de esta manera liberarlo.

Debido a las diversas metodologías usadas en los proyectos de ambos centros, la propuesta estará sustentada en los niveles de pruebas definidos por RUP, con el objetivo de estandarizar la misma.

El uso de un modelo de calidad como la NC ISO/IEC 9126-1:2005, garantiza que los productos, sean evaluados en correspondencia con las características que define dicha norma, garantizando de esta manera, la total calidad de los productos y que se

sienten las bases para lograr que el software esté a la altura de las exigencias de la UCI y del mercado mundial.

CAPÍTULO 2

Diseño del procedimiento de pruebas de liberación para los productos de software de los centros de la facultad 1, basado en las especificaciones de la NC ISO/IEC 9126-1: 2005

Durante el desarrollo del siguiente capítulo, se propone como parte de la solución de la investigación, un procedimiento que tiene como objetivo trazar las pautas a seguir en el proceso de pruebas de liberación de los productos, basándose en la NC ISO/IEC 9126-1:2005, la evaluación de la conformidad de los productos en aras de lograr un alto nivel entre la versión implementada que va a pasar al LIPS.

Actualmente es responsabilidad de los proyectos que pertenecen a los centros de la facultad 1, realizar diferentes tipos de pruebas a los productos que van desarrollando, para encontrar los errores que pudiesen existir durante la implementación. Estas pruebas son realizadas independientemente de la metodología usada y es realizada generalmente por los desarrolladores.

Para lograr que este llegue al LIPS con la menor cantidad de defectos, se hace necesario realizar un procedimiento para guiar los pasos para realizar las pruebas de liberación por los centros, y que estén encaminadas a establecer actividades con el objetivo de liberar el producto. Con vistas a garantizar que estas pruebas se realicen con todas las posibles variantes y que estén guiadas por un modelo de calidad, la propuesta se sustenta en la NC ISO/IEC 9126-1:2005.

Para el diseño, se realiza un análisis crítico de las pruebas que serán desarrolladas y su aplicación en los centros. Este procedimiento, incluye en su propuesta, los requisitos de implantación, los pasos a seguir, así como las plantillas a cumplimentar durante la aplicación del mismo.

El procedimiento se encuentra enfocado a guiar los pasos para que los proyectos realicen las pruebas de liberación para sus productos, para que estos lleguen al LIPS con la menor cantidad de defectos posibles. También establece dos fases a realizar durante el proceso de pruebas, la primera a partir de la evaluación de las

características del software y la segunda, de la identificación de las no conformidades. A continuación se presenta la propuesta del procedimiento para realizar las pruebas de liberación en los centros de la facultad.

2.1 Propuesta del procedimiento de pruebas de liberación para los centros de la facultad 1

1. Objetivo

Describir la estrategia de pruebas de liberación que serán realizadas para los proyectos de los centros de la facultad 1.

2. Alcance

- El Centro de Identificación y Seguridad Digital (CISED) y el Informatización Universitaria (CENIA) pertenecientes a la facultad 1.
- Concierna a todo el personal probador que va a participar en la revisión del software.

3. Definiciones y acrónimos y abreviaturas

Área de prueba: está compuesta por personal inexperto y que desconozca el tema de pruebas, de esta forma, se evalúa que la documentación entregada sea de calidad, que los procesos descritos son tan claros que cualquiera puede entenderlos y el software hace las cosas tal y como están descritas o conformada por programadores con experiencia, personas que saben sin mayores indicaciones en qué condiciones puede fallar una aplicación y que pueden poner atención en detalles que personal inexperto no consideraría.

Casos de pruebas: instrucciones documentadas para la persona que realiza las pruebas que especifican cómo se tiene o tendría que probar una función o una combinación de funciones.

Cliente: representante de un proyecto productivo que solicite los servicios de revisión de un software al laboratorio de calidad.

Conformidad: capacidad del software para adherirse a las normas que se le apliquen o convenciones que se relacionan con la portabilidad.

Defectos del producto: defectos existentes en el producto que hacen que el mismo no cumpla con los requisitos especificados por el cliente para su uso o exista un incumplimiento de los estándares definidos por el grupo de calidad.

Revisión del producto: capacidad para soportar cambios. Facilidad de mantenimiento y esfuerzo requerido para localizar y reparar errores.

4. Responsabilidades de los participantes en el proceso de pruebas (Anexo 1)

Dentro del proceso de pruebas participan dos roles fundamentales: el jefe de pruebas y el probador. Estos roles no son los únicos involucrados durante el proceso de pruebas, aunque para este procedimiento si los más importantes.

También se deben tener en cuenta la participación como observadores dentro del proceso de pruebas del jefe del proyecto, el equipo de desarrollo y el jefe del centro. Estos roles deben estar al tanto del comportamiento del avance de las pruebas que se haga en alguno de los proyectos, pero de manera directa no intervienen en el procedimiento.

En caso de que el centro o el proyecto específicamente no tengan definido un grupo de probadores, es necesario que el asesor de calidad cree un pequeño equipo que sea ajeno a los desarrollados, para así garantizar que el mismo ejecutará pruebas que arrojarán mayores no conformidades.

Se debe garantizar que este equipo, tenga las responsabilidades y competencias que se describen a continuación, y que previamente se encuentre capacitado en numerosos temas relacionados con la calidad del software.

A continuación se describen las responsabilidades del Jefe de pruebas y del probador.

Jefe de pruebas:

1. Experiencia en el proceso de desarrollo de software.
2. Dominio del software que se va a probar.
3. Experiencia en asignar y planear los recursos.
4. Habilidades para el análisis de riesgos, dependencias y toma de decisiones.
5. Habilidades para negociar y comunicar.
6. Conocimiento sobre la administración de proyectos.
7. Capacidad para atacar y resolver los problemas que se presenten.
8. Experiencia en implementar la estrategia de prueba a seguir y elaborar el plan de pruebas.
9. Debe evaluar y documentar los resultados de las pruebas realizadas al software.

10. Debe evaluar el proceso de pruebas y los resultados de las mismas.
11. Debe controlar, monitorear y ejecutar el plan de pruebas y el cronograma establecido.
12. Debe supervisar las pruebas realizadas en las sesiones de trabajo.
13. Recoge todas las no conformidades encontradas y elabora un informe con las mismas que es entregado diariamente al jefe del proyecto.
14. Elabora un informe que es entregado al jefe del proyecto y donde se recoge el resultado de la evaluación de la conformidad del software.

Probador:

1. Realiza las pruebas definidas al software.
2. Recoge las no conformidades detectadas en la plantilla registro de no conformidades
3. Debe tener la habilidad para documentar los errores con claridad y precisión, para correcto entendimiento del equipo de desarrollo.
4. Debe estar capacitado para poder realizar las pruebas eficientemente y alcanzar un alto grado de detención temprana de no conformidades.
5. Debe tener dominio de la ingeniería del software.
6. Ejecuta los diseños de casos de pruebas.
7. Prueba el software a través de las herramientas establecidas para evaluar el mismo con respecto a las siguientes características: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad, portabilidad.

5. Exigencias del equipamiento para las pruebas de liberación

Se debe simular lo mejor posible el ambiente del software para el equipo probador, para garantizar la búsqueda de defectos para su posterior liberación en dependencia de las necesidades de pruebas que exige cada producto. Se debe simular el entorno que requiere por naturaleza, los requisitos y características del software y hardware que necesite el producto en cuestión especificada en el expediente del mismo, de acuerdo con la infraestructura de este.

- PC: Estas deben tener instalados como sistemas operativos: Windows versión XP y Linux Debian.
- Lectores de CD y DVD de 24x.
- Quemador.
- Mouse

- Servidores, suficientemente potente con amplia capacidad de almacenamiento.
- Conexión de red.
- Impresoras.
- Scanner
- Mobiliario para cada participante en el proceso de prueba.
- Mesa con sillas (en dependencia de la cantidad de personas a participar), para el desarrollo de reuniones.
- Local para realizar el adiestramiento del equipo probador que serán los usuarios en este momento (solo se designará en caso de ser necesario).
- Pizarra.
- Plumones.
- Papel.
- Tóner.
- Expediente de proyecto
- Carpetas con las planillas de las incidencias y las no conformidades.
- Carpeta de procedimientos.
- Teléfono.

6. Exigencias del personal participante en las pruebas

Es necesario que los miembros del equipo que van a probar el software tengan algún conocimiento de manera que sea mejor explotar las competencias de los mismos.

Entre las exigencias deben estar:

- Tener conocimiento de la asignatura Ingeniería de Software.
- Debe tener conocimiento y experiencia de cómo se comporta el proceso de desarrollo de software.
- Deben conocer y estar capacitados por parte de los desarrolladores sobre el comportamiento y nivel de acceso al producto.
- En el caso de los probadores, tener experiencia en la administración de proyectos, habilidad para documentar los errores con claridad y precisión.
- Deben ser personas capaces de tener habilidad e interés para aprender nuevas herramientas y tecnologías.

Por parte de la Dirección del Proyecto se brindan cursos de capacitación y nivelación incrementando de esta forma sus conocimientos sobre el tema de calidad del software.

Esta superación puede ser recibida antes o durante del proceso que les permita enfrentarse al problema en cuestión con una visión amplia y con un número mayor de herramientas.

El proyecto que entregue un software o módulo a revisar debe garantizar la capacitación al equipo de probadores, con el objetivo de que estos tengan un amplio dominio del producto y de los niveles de accesos de los roles, para de esta manera asegurar la correcta búsqueda e interpretación de los errores detectados.

7. Planificación y preparación de las pruebas de liberación

Para comenzar el proceso de pruebas es importante planificar las mismas y que el equipo probador esté preparado y capacitado por el proyecto sobre el producto que van a explotar, para garantizar que se pueden recorrer y evaluar todas las funcionalidades, todos los roles y sus niveles definidos. Para la correcta planificación se establecen una serie de actividades que se presentan a continuación.

7.1. Actividades de entrada del producto para las pruebas de liberación.

1. Se debe recibir una solicitud de revisión del producto de software por parte del proyecto, donde se pide satisfacer una necesidad específica, a través del formulario de solicitud de revisión de producto de software. (Anexo 2)
2. Se realiza la reunión de inicio donde participa el jefe de proyecto, el jefe de pruebas, con el objetivo de planificar cómo se le van a realizar las pruebas a este producto, aclarar todas las dudas que puedan existir sobre el proceso de pruebas y definir la fecha de inicio de la primera iteración.
3. En la reunión se planifican las posibles iteraciones y fechas de inicio y fin en el cronograma. (Anexo 3)
4. El líder de proyecto debe entregar todos los artefactos que son obligatorios y necesarios para la realización de las pruebas como: documentación, esta estará en correspondencia con los artefactos que usen según la metodología por la que se guíe el proyecto, pero no puede faltar el manual de usuario; software y los diseños de casos de pruebas con los casos de usos correspondientes.

5. Debe quedar claramente definido el tipo de software que será recibido, su complejidad, además de exigir que este sea estable, es decir, en caso todo debe estar en su versión beta.
6. Se debe hacer un chequeo del producto entregable para así exigir que se entreguen documentos tales como manual de usuario, glosario de términos, descripción de los casos de uso del sistema, especificación de requisitos del software y diseño de casos de prueba, entre otros que sean necesarios.
7. Una vez establecido todos los elementos y entregables, se firma el acta de entrega.
8. Debe realizarse una estimación del tiempo y tamaño del proyecto que tribute a la cantidad de casos de uso, de personas que se necesitan para revisar el proyecto. Además, que con estos casos de usos obtener una clasificación para considerar las variables tales como: complejidad del mismo, cantidad de requisitos, experiencia que se tenga en su solución, relevancia en la arquitectura, jerarquía dentro de esta, etc.
9. La calidad del producto del software se debe evaluar usando un modelo de calidad definido. El modelo de calidad se usará al fijar los objetivos de calidad para los productos del software y productos de software intermedios. (Anexo 4)
10. Según el tipo de proyecto que se va a revisar se determinará el objetivo de las pruebas, un estimado de las que se les va a aplicar y las herramientas a utilizar.
11. Se debe inspeccionar el producto y que estos elementos queden recogidos en la plantilla resultado de la inspección del producto y acta de entrega. (Anexo 5 y Anexo 6)
12. Una vez conciliada la entrega por ambas partes se firma el acta de aceptación del producto. (Anexo 7)

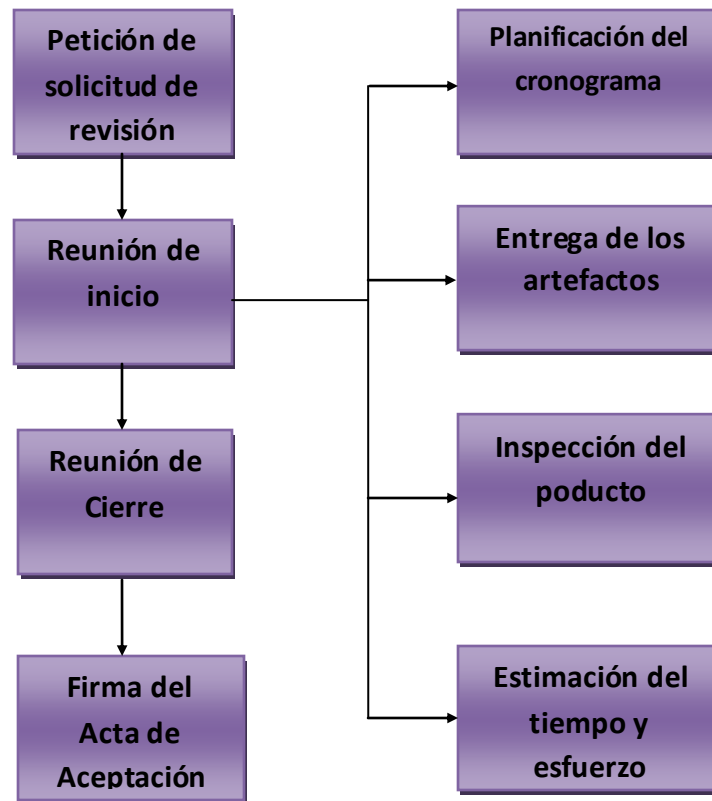


Fig. 2.1. Descripción gráfica de las actividades de entrada para la liberación

7.2. Estrategia para realizar las pruebas de liberación

A partir de las necesidades de los centros se define una estrategia fácil de implementar. Esta estrategia estará enfocada a dos fases principales:

- Fase 1: Probar la funcionalidad del software para identificar las no conformidades.
- Fase 2: Evaluar la conformidad del software en las seis características definidas en la ISO/IEC 9126-1:2005.

De manera general los roles participantes en el desarrollo de estas actividades son:

- Jefe de pruebas
- Probador

Al frente del equipo probador se encuentra el jefe de pruebas, como máxima figura y guía y como encargado de que se lleven de manera adecuada y completa las dos fases definidas anteriormente, con el objetivo de alcanzar un producto evaluado que tenga la mayor calidad posible.

Esta estrategia puede redefinirse si algún producto específico lo requiere debido a las características que pueda presentar o a la necesidad inmediata que pueda surgir.

Para la liberación del producto se va a establecer un índice de valores que van a estar dados por los pesos establecidos para evaluar que el producto cumpla con cada una de las 6 características.

Para esto se diseña una plantilla que contiene los pesos establecidos y su significado para calcular el puntaje del producto. Para esto es necesario:

- Realizar primero los tipos de pruebas que se describen en el acápite siguiente, para evaluar el producto en cada uno de estas características y sus sub características.
- Establecer el peso en la plantilla de evaluación de los pesos por las características (Anexo 11), el peso estará en correspondencia con el resultado de la prueba realizada.
- Analizar el puntaje total para establecer si el producto está conforme o no.
- Documentar los resultados satisfactorios e insatisfactorios de la evaluación en la plantilla informe de los resultados de la evaluación de la conformidad del software (Anexo 12). El mismo es realizado por el jefe de pruebas y entregado al jefe del proyecto como parte de la liberación del producto.
- En caso de detectar no conformidades se recogen en el registro de no conformidades, y el jefe de pruebas elabora el informe y se la entrega al jefe del proyecto para que el equipo de desarrollo las vaya solucionando.

Una vez evaluado el producto, el equipo de probador pasa a identificar las no conformidades.

7.7.1. Probar la funcionalidad del software para identificar las no conformidades

Es recomendable tener algunas actividades fundamentales en cuenta para la identificación de las no conformidades. Para esto se realizan pruebas de caja negra para verificar la funcionalidad del producto. Estas actividades están encaminadas a:

- El probador realiza pruebas exploratorias al software con el objetivo de encontrar no conformidades desde el punto de vista funcional, del diseño e inconsistencias en los requerimientos que fueron implementados.

- Los probadores ejecutarán los diseños de casos de pruebas con el juego de datos anteriormente definido por el equipo de desarrollo para probar los posibles escenarios con datos válidos e inválidos. En caso de detectar no conformidades estas serán recogidas.
- El probador realizará pruebas a la documentación: manual de usuario, descripción de los casos de uso del sistema, etc.
- Las no conformidades detectadas serán enviadas diariamente por el jefe de pruebas al jefe del proyecto para que el equipo de desarrollo le dé respuesta a las mismas.
- Una vez terminadas las pruebas exploratorias se realizarán las pruebas a los casos de pruebas para validar los posibles escenarios para los casos de uso.
- Se realizarán tantas iteraciones como sean necesarias con el objetivo de ir disminuyendo las no conformidades encontradas.
- Una vez que no existan no conformidades en el software y haya quedado establecida la constancia y el porqué de aquellas que no fueron resueltas por el equipo de desarrollo, entonces se podrá dar fin a las pruebas de liberación.
- Se realizará una reunión de cierre donde los acuerdos son recogidos en la minuta de reunión y se firma por ambas partes el acta de aceptación.

7.7.2. Evaluar la conformidad del software en las 6 características definidas en la ISO/IEC 9126-1:2005

La NC ISO/IEC 9126-1:2005 especifica las principales características y sub características que debe cumplir un software cuando se evalúa para que esté conforme. Para cada una de estas características, existen en la actualidad numerosas herramientas automatizadas, que serán utilizadas por los probadores durante el proceso de pruebas de liberación. El uso de ellas permitirá probar que el producto cumple con las características propuestas e identificar los errores que se producen y así eliminarlos antes de ser liberado. (ISO-9000-2005)

Es importante aclarar que a medida que se vayan realizando los diferentes tipos de pruebas, en correspondencia con el resultado de la misma, se dará un puntaje que será recogido en la plantilla de evaluación de los pesos por las características.

Una vez terminadas todas las pruebas y establecido un peso el resultado del puntaje es recogido en el informe de los resultados de la evaluación de la conformidad del software.

A continuación se guían cada uno de los tipos de pruebas que se van a realizar y que tributan a la evaluación de las características, y se recomiendan las herramientas a utilizar.

Las no conformidades que se encuentren en la realización de las mismas serán recogidas en el registro de no conformidades, donde se debe hacer una para cada tipo de prueba que se esté realizando.

7.7.2.1. Tipos de pruebas a realizar (Cruz, 2008)

Con el objetivo de analizar si el software cumple con las seis características definidas en la norma, se deben realizar diferentes tipos de pruebas, que al ejecutarlas permitirá conocer si el producto cumple o no con alguna de ellas, pues será el propósito que las guían.

Con vistas a evaluar los productos de los centros de la facultad 1, en la investigación se proponen diferentes pruebas que deben ser ejecutadas después que acabe la anterior.

Las pruebas que se proponen son:

1. Prueba de funcionalidad
2. Prueba de seguridad y control de acceso
3. Prueba de volumen
4. Prueba de usabilidad
5. Prueba de integración de los datos y la base de datos
6. Prueba de estructura
7. Prueba de interfaz de usuario
8. Prueba de performance (tiempo de respuesta)
9. Prueba de benchmark (comparador de rendimiento)
10. Prueba de carga
11. Prueba de esfuerzo
12. Prueba de fallas y recuperación

Cada una de estas pruebas está enfocada a evaluar elementos diferentes, y se deben ejecutar en el orden que son mencionadas. Estas pruebas son flexibles, es decir, si el proyecto considera que por las características que presenta su producto, puede

eliminarse alguna, esta debe quedar recogida en la reunión de inicio cuando se traza la estrategia para llevar a cabo el proceso de pruebas de liberación.

Los probadores deben analizar las características del producto y chequear lo acordado en la reunión de inicio, por si no es necesario realizar otro tipo de prueba que no haya quedado acordada, en ese caso, se debe recoger en el informe las causas de las características que no fueron probadas.

Para la evaluación de la conformidad del software se proponen herramientas automatizadas para llevarlas a cabo. En algunos casos, que es más de una, el equipo probador define cuál utilizará en correspondencia con las características del producto que vaya a evaluar. A continuación se establecen los elementos fundamentales que persigue cada una de las pruebas y hacia qué está enfocada, así como las herramientas posibles a utilizar.

1. Prueba de funcionalidad

Prueba centrada en validar las funciones que son objeto de prueba como lo que deben ser, ofreciendo los servicios, métodos o casos de usos requeridos. Esta prueba es implementada y ejecutada contra diferentes objetos de pruebas, incluyendo unidades, unidades integradas, aplicaciones y sistemas. Su objetivo es verificar la aceptación de los datos, el proceso, la recuperación y la implementación correcta de las reglas del negocio. Durante la ejecución de esta prueba se deben usar datos válidos e inválidos, para verificar que:

- Se obtienen los resultados esperados cuando se usan datos válidos.
- Cuando se usan datos inválidos se despliegan los mensajes de error o advertencia apropiados.
- Se aplica apropiadamente cada regla del negocio.

Esta prueba se les aplica a los instaladores, a la aplicación y al diseño de casos de pruebas. Al realizar esta prueba se debe comprobar:

- La idoneidad (**subcaracterística de la ISO 9126**): del sistema, porque el software debe ser capaz de mantener un juego apropiado de funciones para las tareas especificadas y los objetivos del usuario.
- La precisión (**subcaracterística de la ISO 9126**): capacidad del software para proporcionar efectos o resultados correctos o convenidos.
- La conformidad (**subcaracterística de la ISO 9126**): capacidad del software para adherirse a las normas que se le apliquen, convenciones, regulaciones, leyes y las prescripciones similares.

Herramientas que se usan para esta prueba

JMeter: puede ser utilizado como una herramienta para analizar y medir el desempeño de una variedad de servicios. Además, posee la capacidad de realizar desde una solicitud sencilla hasta secuencias de peticiones que permiten diagnosticar el comportamiento de una aplicación en condiciones de producción.

VTest: herramienta de prueba funcional y de la regresión automatizada para los usos de la tela. Incorpora el expediente, lo verifica, aparato de lectura y las capacidades de la divulgación. No requiere un fondo de programación. Para esos usuarios que deseen utilizar scripting, utiliza JavaScript como la lengua scripting.

Selenium: herramienta para realización de pruebas de aplicaciones web. Está orientado a la ejecución de pruebas funcionales a nivel de usuario directamente desde el navegador. Permite desarrollar scripts para pruebas de aplicaciones Web en diversos lenguajes como Java, Ruby, Python, Perl, .Net o PHP. Automatiza el proceso de pruebas y permite ejecutar un conjunto de pruebas completo si es necesario o pruebas particulares.

TestPartner de compuware: herramienta que automatiza las pruebas funcionales y de regresión. Ha sido especialmente diseñada para complejas aplicaciones basadas en Microsoft, Java y tecnologías web.

Webking de parasoft: herramienta de Parasoft, para automatizar las pruebas de las aplicaciones web (análisis de riesgos de los sitios, pruebas funcionales, pruebas de carga, rendimiento y análisis de seguridad).

Checking de ALS: herramienta de monitorización del proceso de desarrollo software y sus resultados, que cubre las necesidades de organizaciones que desean controlar la calidad del software antes de su puesta en producción.

Jameleon: herramienta para probar aplicaciones de forma automática. Divide la aplicación en características y crea casos de prueba para cada una de ellas. Permite reducir enormemente el costo de crear Testcases (Pruebas funcionales o unitarias) que manejan librerías tales como: Junit, httpUnit, etc. Además, puede probar aplicaciones web de todo tipo.

2. Prueba de seguridad y control de acceso

Esta prueba se enfoca en dos áreas de seguridad:

- Seguridad en el ámbito de aplicación, incluyendo el acceso a los datos y a las funciones de negocios. Verifica que un actor pueda acceder solo a las funciones o datos para los cuales su tipo de usuario tiene permiso. Para ello se debe

identificar y hacer una lista de cada tipo de usuario y las funciones y datos sobre las que cada tipo tiene permiso; crear pruebas para cada tipo de usuario y verificar cada permiso creando operaciones específicas para cada tipo de usuario y modificar el tipo de usuario y volver a ejecutar las pruebas para los mismos usuarios. En cada caso, verificar que las funciones o datos adicionales están correctamente disponibles o son denegados.

- Seguridad en el ámbito de sistema, incluyendo conexión, o acceso remoto al sistema. Verificar que solo los actores con acceso al sistema y a las aplicaciones, puedan acceder a ellos. Para esto el acceso al sistema debe ser discutido con el administrador del sistema o la red. Esta prueba no puede requerirse como tal, es una función del administrador del sistema o de la red.

Al realizar esta prueba se debe comprobar:

- La capacidad del software para proteger información y los datos, para que personas o sistemas desautorizados no puedan leer o pueden modificar los mismos, y a las personas o sistemas autorizados no les sea denegado el acceso a ellos. **Seguridad (Característica de la ISO 9126)**

Herramientas que se usan para esta prueba

Nessus: es un escáner de seguridad remoto para Linux, BSD, Solaris y Otros Unix. Está basado en plug-in(s), tiene una interfaz basada en GTK, y realiza más de 1200 pruebas de seguridad remotas. Permite generar reportes en HTML, XML, LaTeX, y texto ASCII; también sugiere soluciones para los problemas de seguridad.

Ethereal: es un analizador de protocolos de red para Unix y Windows, y es libre. Permite examinar datos de una red viva o de un archivo de captura en algún disco. Se puede examinar interactivamente la información capturada, viendo información de detalles y sumarios por cada paquete. Tiene varias características poderosas, incluyendo un completo lenguaje para filtrar lo que se quiera ver y la habilidad de mostrar el flujo reconstruido de una sesión.

Snort: sistema de detección de intrusiones de red de poco peso (para el sistema), capaz de realizar análisis de tráfico en tiempo real y registro de paquetes en redes con IP. Puede realizar análisis de protocolos, búsqueda/identificación de contenido y puede ser utilizado para detectar una gran variedad de ataques y pruebas. Utiliza un lenguaje flexible basado en reglas para describir el tráfico que debería recolectar o dejar pasar, y un motor de detección modular.

Fortify Security Tester: para Visual Studio 2005 Team System, proporciona las pruebas de seguridad eficaces a los equipos de desarrollo, permitiéndoles verificar la adecuación a los estándares de seguridad, y posibles vulnerabilidades en el código de sus aplicaciones antes de su despliegue.

3. Prueba de volumen

Centrada en verificar las habilidades de los objetos de prueba para manejar grandes cantidades de datos para determinar si se alcanzan límites que causen la falla del software, tanto en entrada como en salida, o residente en la base de datos.

- Máximo (real o físicamente posible) número de clientes conectados, o simulados, todos realizando la misma operación (peor caso de operación) por un período de tiempo extenso.
- Máximo tamaño de base de datos y múltiples consultas ejecutadas simultáneamente.

La prueba de volumen identifica la carga máxima continua que puede manejar el software a prueba en un período dado. Puede incluir un procedimiento que indique el uso de consultas que devuelvan todo el contenido de la base de datos, o cuando la cantidad de datos de entrada excede a la cantidad establecida de cada campo.

Se deben usar múltiples clientes, ejecutando las mismas pruebas o pruebas complementarias para producir el peor caso de volumen de operaciones o mezcla en un período de tiempo extenso. Se debe crear el tamaño máximo de base de datos (real, escalado o con datos representativos) y múltiples clientes ejecutando consultas simultáneamente por un período de tiempo extenso. Al realizar esta prueba se debe comprobar:

- La conformidad (**subcaracterística de la ISO 9126**): capacidad del software para adherirse a las normas que se le apliquen, convenciones, regulaciones, leyes y las prescripciones similares.
- La madurez (**subcaracterística de la ISO 9126**): capacidad del software de evitar una avería como resultado de haberse producido un fallo del software.
Confiabilidad (Característica de la ISO 9126).
- La tolerancia ante fallos (**subcaracterística de la ISO 9126**): capacidad del software de mantener un nivel de ejecución específico en caso de fallos del

software o de infracción de sus interfaces especificadas. **Confiabilidad (Característica de la ISO 9126).**

- La utilización de recursos (**subcaracterística de la ISO 9126**): capacidad del software para usar los recursos apropiados en un plazo de tiempo adecuado cuando el software realiza su función bajo las condiciones declaradas. **Eficiencia (Característica de la ISO 9126).**

Herramientas que se usan para esta prueba

VTest

JUnit

Es un framework para realizar y automatizar pruebas de aplicaciones Java. Incorpora las ventajas tales como: los métodos que representan las pruebas a realizar ya no tienen que tener el prefijo "test" (prueba) en su nombre sino que se indican con la anotación @Test .Se pueden realizar pruebas en donde el tiempo de ejecución es crítico. De manera que una prueba falle si tarda más de x milisegundos en ser ejecutada. Se pueden realizar pruebas en donde se debe controlar que una excepción es lanzada.

Data Generator: es un generador automático de datos de prueba para pruebas de bases de datos a gran escala y garantiza la calidad.

4. Prueba de usabilidad

Prueba encaminada a factores humanos, estéticos, consistencia en la interfaz de usuario, ayuda sensitiva al contexto y en línea, asistente documentación de usuarios y materiales de entrenamiento. Al realizar esta prueba se debe comprobar:

- **La comprensibilidad (subcaracterística de la ISO 9126):** es la capacidad del producto de software para permitirle al usuario entender si el software es conveniente, y cómo puede usarse para las tareas particulares y condiciones de uso.
- **La operabilidad (subcaracterística de la ISO 9126):** es la capacidad del producto del software para permitirle al usuario aprender de su aplicación.
- **La cognoscitividad (subcaracterística de la ISO 9126):** es la capacidad del producto del software para permitirle al usuario aprender su aplicación.

- **La operabilidad (subcaracterística de la ISO 9126):** es la capacidad del producto del software para permitirle al usuario operarlo y controlarlo.
- **La atracción (subcaracterística de la ISO 9126):** es la capacidad del producto del software de ser amigable para el usuario.

Herramientas que se usan para esta prueba

Selenium

5. Prueba de integración de los datos y la base de datos

Está enfocada a la valoración de la robustez, es decir a la resistencia a fallos. Asegura que los métodos y procesos de acceso a la base de datos funcionen correctamente y sin corromper datos. Se debe invocar a cada método o proceso de acceso a la base de datos con datos válidos e inválidos e inspeccionar la base de datos para asegurarse de que se han guardado los datos correctos; que todos los eventos de la base de datos ocurrieron correctamente, o los datos devueltos para asegurar que se recuperaron datos correctos por la vía correcta.

Se deben usar bases de datos pequeñas para aumentar la facilidad de inspección de los datos para verificar que no sucedan eventos no aceptables. Al realizar esta prueba se debe comprobar:

- **La madurez (subcaracterística de la ISO 9126):** es la capacidad del software de evitar una avería como resultado de haberse producido un fallo del software. **Confiabilidad (Característica de la ISO 9126).**
- **La tolerancia ante fallos (subcaracterística de la ISO 9126):** es la capacidad del software de mantener un nivel de ejecución específico en caso de fallos del software o de infracción de sus interfaces especificadas. **Confiabilidad (Característica de la ISO 9126).**
- **La restaurabilidad (subcaracterística de la ISO 9126):** es la capacidad del software de restablecer su nivel de ejecución y recobrar los datos directamente afectados en caso de avería. **Confiabilidad (Característica de la ISO 9126).**
- **La utilización de recursos (subcaracterística de la ISO 9126):** es la capacidad del software para usar los recursos apropiados en un plazo de tiempo adecuado cuando el software realiza su función bajo las condiciones declaradas. **Eficiencia (Característica de la ISO 9126).**

- **La diagnósticabilidad (subcaracterística de la ISO 9126):** la capacidad del producto del software ser diagnosticado para detectar deficiencias o causas de fracasos o averías en el software, o para las partes para ser modificadas para ser identificadas. **Mantenibilidad (Característica de la ISO 9126).**
- **La conformidad (subcaracterística de la ISO 9126):** es la capacidad del software para adherirse a las normas que se le apliquen, convenciones, regulaciones, leyes y las prescripciones similares. **Confiabilidad (Característica de la ISO 9126).**

Herramientas que se usan para esta prueba

CheckKing de ALS: herramienta de monitorización del proceso de desarrollo software y sus resultados, que cubre las necesidades de organizaciones que desean controlar la calidad del software antes de su puesta en producción.

TrackRecord de Compuware: se ajusta a cualquier proceso de desarrollo y pruebas, ofreciendo un sistema de rastreo que ayuda en la identificación y resolución de defectos software.

6. Prueba de estructura

Está enfocada a la valoración a la adherencia a su diseño y formación. Este tipo de prueba es hecho a las aplicaciones Web asegurando que todos los enlaces están conectados, el contenido deseado es mostrado y no hay contenido huérfano. Al realizar esta prueba se debe comprobar:

- **La restaurabilidad (subcaracterística de la ISO 9126):** es la capacidad del software de restablecer su nivel de ejecución y recobrar los datos directamente afectados en caso de avería. **Confiabilidad (Característica de la ISO 9126).**
- **La utilización de recursos (subcaracterística de la ISO 9126):** es la capacidad del software para usar los recursos apropiados en un plazo de tiempo adecuado cuando el software realiza su función bajo las condiciones declaradas. **Eficiencia (Característica de la ISO 9126).**
- **La conformidad (subcaracterística de la ISO 9126):** es la capacidad del software para adherirse a las normas que se le apliquen, convenciones,

regulaciones, leyes y las prescripciones similares. **Confiablez**
(Característica de la ISO 9126).

Herramientas que se usan para esta prueba

VTest

7. Prueba de interfaz de usuario

Con esta prueba se verifica que la interfaz de usuario proporcione al usuario el acceso y navegación, a través de las funciones apropiada, es decir, que la navegación a través de los elementos que se están probando reflejen las funciones del negocio y los requerimientos, incluyendo manejo de ventanas, campos y métodos de acceso; los objetos de las ventanas y características, como menú, tamaño, posición, y estado funcionen de acuerdo con los estándares.

Además, asegura que los objetos presentes en la interfaz de usuario se muestren como se espera y conforme a los estándares establecidos.

Herramientas que se usan para esta prueba

Para esta prueba serán utilizadas las listas de chequeo.

8. Prueba de performance (tiempo de respuesta)

En esta prueba se miden y evalúan los tiempos de respuesta, los tiempos de transacción y otros requerimientos sensitivos al tiempo. El objetivo de la prueba es verificar que se logren los requerimientos de performance.

La prueba de performance es implementada y ejecutada para poner a punto los destinos de pruebas de performance como función de condiciones de trabajo o configuraciones de hardware.

Para ello se debe:

- a) Usar procedimientos de prueba desarrollados para verificar funciones o ciclos de negocio.
- b) Modificar archivos de datos para aumentar el número de transacciones o los procedimientos de prueba para aumentar el número de iteraciones de ocurrencia de transacciones.
- c) Las pruebas se deben ejecutar en una máquina y se debe repetir con múltiples usuarios (virtuales o reales).

Las pruebas de performance deben incluir un trabajo de fondo en el servidor. Esto se puede realizar de distintas formas:

- Enviar transacciones directamente al servidor, generalmente en la forma de consultas (SQL).
- Crear usuarios virtuales para simular muchos clientes, generalmente varios cientos. Se pueden usar herramientas de Emulación de Terminar Remota para lograr este objetivo. Esta técnica también se usa para cargar la red con “tráfico”.
- Usar muchos clientes físicos, cada uno corriendo procedimientos de prueba.

La prueba de performance se debe realizar en una máquina dedicada para permitir control total y medición exacta.

Las bases de datos usadas para las pruebas de performance deben tener un tamaño similar a las reales. Al realizar esta prueba se debe comprobar:

- El crono-comportamiento (**subcaracterística de la ISO 9126**): la capacidad del software para proporcionar una respuesta apropiada y los tiempos de procesamiento al realizar su función, bajo condiciones declaradas.
- La conformidad (**subcaracterística de la ISO 9126**): la capacidad del software para adherirse a las normas que se le apliquen o convenciones que se relacionan con la eficiencia.

Herramientas que se usan para este tipo de prueba

IPS Performance Optimizer de Hyperformix: herramienta centrada en los entornos de preproducción, que proporciona la garantía del rendimiento de principio a fin de aplicaciones distribuidas complejas.

QACenter Performance Edition de Compuware: ayuda a realizar pruebas realistas, con profundidad y reproducibles para aplicaciones de *e-business* (comercio electrónico), ERP y cliente/servidor. Gracias a una potente combinación de herramientas para pruebas de carga, gestión de datos y monitorización de servidores, QACenter Performance Edition permite a las empresas buscar y resolver rápidamente problemas de rendimiento.

9. Prueba de benchmark (comparador de rendimiento)

Compara el rendimiento de un elemento nuevo o desconocido a uno de carga de trabajo de referencia conocido. Al realizar esta prueba se debe comprobar:

- La utilización de recursos (**subcaracterística de la ISO 9126**): la capacidad del software para usar los recursos apropiados en un plazo de tiempo adecuado cuando el software realiza su función bajo las condiciones declaradas.
- La conformidad (**subcaracterística de la ISO 9126**): la capacidad del software para adherirse a las normas que se le apliquen o convenciones que se relacionan con la eficiencia.

Herramientas que se usan para este tipo de prueba

Benchmark Factory: herramienta de rendimiento altamente escalable, capaz de simular el acceso de miles de usuarios a sus servidores de bases de datos, archivos, Internet y correo. Permite determinar la capacidad del sistema, localizar los posibles cuellos de botella y aislar problemas relacionados con sobrecargas del sistema dentro de su entorno informático distribuido.

10. Prueba de carga

La prueba de carga somete los objetos a verificar a diferentes cargas de trabajo para medir y evaluar los comportamientos de tiempo de respuesta y la habilidad de los objetos de continuar funcionando apropiadamente bajo diferentes cargas de trabajo. El objetivo es determinar y asegurar que el sistema funciona apropiadamente en circunstancias de máxima carga de trabajo esperada. Además, de evaluar las características de tiempo de respuesta, como tiempos de respuesta, tiempos de transacciones y otros elementos sensitivos al tiempo.

Es importante señalar que la prueba de carga debe realizarse en una máquina dedicada para tener control total y exactitud de mediciones, y las bases de datos usadas para la prueba deben tener un tamaño similar a las reales. Al realizar esta prueba se debe comprobar:

- La utilización de recursos (**subcaracterística de la ISO 9126**): la capacidad del software para usar los recursos apropiados en un plazo de tiempo adecuado cuando el software realiza su función bajo las condiciones declaradas.
- La conformidad (**subcaracterística de la ISO 9126**): la capacidad del software para adherirse a las normas que se le apliquen o convenciones que se relacionan con la eficiencia.

Herramienta que se usa en esta prueba

Jmeter, QACenter Performance Edition de Compuware

QALoad de Compuware: herramienta de pruebas de carga que ayuda a los equipos de pruebas, desarrolladores y jefes de proyecto a realizar pruebas de cargas efectivas a aplicaciones distribuidas.

SOATest de Parasoft: herramienta que proporciona las pruebas y verificación instantáneas de Servicios Web, simplifica los desarrollos **SOA** automatiza las pruebas funcionales cliente/servidor, pruebas de regresión, pruebas de carga y rendimiento, etc.

Webking de Parasoft: herramienta para automatizar las pruebas de las aplicaciones Web (análisis de riesgos de los sitios, pruebas funcionales, pruebas de carga y rendimiento y análisis de seguridad).

11. Prueba de esfuerzo (stress, competencia por recursos, bajos recursos)

La prueba de esfuerzo es un tipo de prueba de performance implementada y ejecutada para encontrar errores cuando hay pocos recursos o cuando hay competencia por recursos. Poca memoria o poco espacio de disco pueden revelar fallas en el software que no aparecen bajo condiciones normales de cantidad de recursos.

Otras fallas pueden resultar al competir por recursos compartidos como bloqueos de bases de datos o ancho de banda de red. La prueba de esfuerzo también puede usarse para identificar el trabajo máximo que el software puede manejar. El objetivo de la prueba de esfuerzo es también identificar y documentar las condiciones bajo las cuales el sistema falla y no continúa funcionando apropiadamente.

Para probar recursos limitados, las pruebas se deben ejecutar en una sola máquina, y se debe reducir o limitar la memoria en el servidor. Para las pruebas de esfuerzo restantes, deber usarse múltiples clientes, cualquiera que ejecute las mismas pruebas o pruebas complementarias para producir el peor caso de volumen de operaciones.

Pueden requerir herramientas de red para cargar la red con mensajes o paquetes. La cantidad de disco del servidor usada por el sistema debe ser reducida temporalmente para restringir el espacio disponible para crecimiento de la base de datos. Sincronizar el acceso simultáneo de varios clientes accediendo a los mismos datos. Al realizar esta prueba se debe comprobar:

- **La restaurabilidad (subcaracterística de la ISO 9126):** es la capacidad del software de restablecer su nivel de ejecución y recobrar los datos directamente afectados en caso de avería. **Confiabilidad (Característica de la ISO 9126).**
- **La utilización de recursos (subcaracterística de la ISO 9126):** es la capacidad del software para usar los recursos apropiados en un plazo de tiempo adecuado cuando el software realiza su función bajo las condiciones declaradas. **Eficiencia (Característica de la ISO 9126).**
- **La conformidad (subcaracterística de la ISO 9126):** es la capacidad del software para adherirse a las normas que se le apliquen, convenciones, regulaciones, leyes y las prescripciones similares. **Confiabilidad (Característica de la ISO 9126).**

Herramienta que se usa en esta prueba

Open Load Tester de Open Demand Systems: es la primera solución rápida de optimización de rendimiento basada en navegador, fácil de usar, para las pruebas de carga y stress de aplicaciones y sitios web dinámicos.

JCrawler: herramienta para realizar pruebas de estrés a aplicaciones web. Es apropiado para portales complejos en los que hay que probar todas las páginas del portal y no solo algunas URLs. Es independiente de la plataforma, posee un modo consola y es sencillo de configurar. Es independiente del sistema operativo que se use.

12. Prueba de fallas y recuperación

Estas pruebas aseguran que el software puede recuperarse de fallas de hardware, software o mal funcionamiento de la red sin pérdida de datos o de integridad de los datos. La prueba de recuperación es un proceso en el cual la aplicación o sistema se expone a condiciones extremas, o condiciones simuladas, para causar falla, como fallas en dispositivos de entrada/salida o punteros inválidos a la base de datos. Los procedimientos de recuperación se invocan y la aplicación o sistema es monitoreado e inspeccionado para verificar que se recupera apropiadamente la aplicación o sistema y se logre la recuperación de datos. En la prueba se incluyen los siguientes tipos de condiciones:

- Interrupción de energía al cliente.

- Interrupción de energía al servidor.
- Interrupción de comunicaciones mediante los servidores de la red.
- Interrupción de comunicación o pérdida de energía de los discos del servidor o con los controladores.
- Ciclos incompletos (procesos de filtro de datos interrumpidos, procesos de sincronización de datos interrumpidos).
- Punteros o claves inválidos a la base de datos.
- Elementos de datos inválidos o corruptos en la base de datos.

Es de suma importancia para la realización de estas pruebas simular las siguientes acciones, individualmente:

1. Interrumpir la energía del cliente: apagar la PC.
2. Interrumpir la energía del servidor: simular o iniciar el proceso de apagado del servidor.
3. Interrupción por medio de los servidores de red: simular o iniciar la pérdida de comunicación con la red (desconectar físicamente la comunicación o apagar el servidor de red).
4. Interrumpir la comunicación o quitar la energía de los discos del servidor o sus controladores: simular o eliminar físicamente la comunicación con uno o más controladores de disco o los discos.

Las últimas dos pruebas requieren que se logre un estado conocido de la base de datos. Se deben corromper manualmente campos de la base de datos, punteros y claves trabajando directamente sobre la base de datos (utilizando herramientas para la base de datos). Se deben ejecutar las pruebas de funcionalidad y verificar que los ciclos se completen.

Los procedimientos para desconectar cables (simulando falta de energía o pérdida de comunicación) no son deseables o factibles. Se pueden requerir métodos alternativos, como software de diagnóstico. Se requieren los grupos de recursos de sistemas, bases de datos y red. Estas pruebas deben ejecutarse fuera del horario de trabajo normal o en una máquina aislada.

Herramientas que se usan para esta prueba

JMeter, Selenium

7.8. Artefactos

7.8.1. Diseño de los casos de pruebas

Para la realización de los casos de pruebas y las pruebas exploratorias se encuentran definidas por Calisoft, las plantillas: diseño de casos de pruebas (Anexo 8) y el registro de no conformidades (Anexo 9) respectivamente. Para el diseño de los casos de pruebas, se encuentra establecido por Calisoft que el equipo desarrollador debe llenar la plantilla en su mayoría. Solo es responsabilidad del equipo probador realizar las pruebas con el juego de datos ya establecidos y entregados por el equipo de desarrollo para la realización de las mismas.

7.8.2. Registro de no conformidades

Las no conformidades encontradas diariamente en las sesiones de trabajo, serán recogidas y guardadas en el expediente de proyecto en correspondencia con la iteración en la que se encuentren las pruebas. Estas se recogerán en la plantilla de registro de no conformidades.

Se emitirán informes diarios que serán enviados al jefe del proyecto o del módulo (puede ser también la persona que se encuentre al frente del software a probar). Estos reportes se recogerán en la plantilla de no conformidades y contendrán todas las no conformidades detectadas en las sesiones de trabajo del día correspondiente.

7.8.3. Registro de evaluación de los pesos por las características

El resultado de la ejecución de los tipos de pruebas para comprobar que el producto cumple con cada una de las características definidas en la norma, será recogido en el registro de evaluaciones de los pesos por las características. Esta plantilla es propuesta para que el probador pueda asignar un peso, en correspondencia con cada una de las características que cumple el producto después de ser evaluado.

El puntaje final de los pesos, determina para las características evaluadas si el producto es conforme o no. El índice de conformidad debe ser igual o mayor a 3.5 puntos.

7.8.4. Informe de los resultados de la evaluación de la conformidad del software

Luego de obtenido el resultado de la evaluación de la conformidad del producto se elabora un informe final por el jefe de pruebas, donde se recogen los elementos principales de la evaluación, el resultado y las causas de los elementos no evaluados. Este informe será entregado al jefe del proyecto como constancia del proceso de pruebas de liberación.

7.8.5. Acta de aceptación

Al finalizar las pruebas de liberación se realizará una reunión entre las personas relacionadas con las pruebas y que participaron en la reunión de entrega de los materiales para el comienzo de las pruebas. Los puntos y acuerdos tratados en la reunión se recogerán en la minuta de reunión (Anexo 10) establecida por Calisoft.

Una vez que las dos partes hayan conciliado y analizado los reportes en los que se encuentra el resultado de las pruebas de liberación, se pasa a firmar el acta de aceptación como constancia de que terminaron las pruebas y de que el software se encuentra listo para ser entregado a Calisoft a que se le realicen pruebas posteriores de liberación a nivel central.

8. Culminación del proceso de pruebas

Cuando las dos fases definidas para realizar las pruebas se culminan, se debe revisar que el producto tenga ya los elementos necesarios y la menor cantidad de no conformidades para pasar a liberarlo. Para determinar si este ya puede ser liberado se deben tener en cuenta los siguientes elementos:

- Se deben haber resuelto la mayor cantidad de no conformidades detectadas. Se puede dar el caso de que quede alguna rechazada o pospuesta pero que no sea de un alto nivel de importancia.
- Se debe haber evaluado el producto, a partir de la asignación del peso luego de haber realizado las diferentes pruebas definidas para evaluar el mismo.
 - ✓ La asignación de los pesos será dada por el probador y estará en correspondencia con el resultado del tipo de prueba ejecutado y con el criterio que crea correcto el probador. Si el índice de los pesos, resultado del puntaje total, tiene un nivel adecuado y que sea mayor que 3.5 puntos, entonces se considera que el producto está evaluado conformemente.
- Se debe haber elaborado el informe donde se recojan los resultados de la evaluación y el informe que recoja las no conformidades detectadas durante las iteraciones realizadas y el resultado de estas.

Una vez que el jefe de pruebas tiene todos estos elementos, se realiza la reunión de cierre donde se revisa con el jefe de proyecto y el director del centro (en caso de ser necesario), los resultados del proceso de pruebas ejecutados.

Si las dos partes están de acuerdo se hace entrega al jefe del proyecto de los informes, se levanta la minuta de reunión, se firma el acta de aceptación y se da por concluido el proceso de pruebas y queda liberado el producto, para pasar al LIPS.

2.2 Conclusiones Parciales

El procedimiento para pruebas de liberación de los productos de los proyectos de los centros de la facultad 1, se propone con el objetivo de establecer una guía para que el probador conozca los pasos a seguir para que los proyectos puedan comenzar a llevar a cabo el proceso de liberación.

Como parte de que los centros alcancen conocimiento en función de lograr un paso importante en la calidad del producto implementado que pasará al LIPS, esta propuesta, a través de plantillas y herramientas que propone para su utilización en los diferentes tipos de pruebas, permitirá aplicar la estrategia definida e ir mejorando la misma, y de manera más importante, asegurará que los proyectos puedan llevar a cabo un proceso tan importante y complejo como es la liberación de sus productos, independientemente de las metodologías que usen.

Validación del procedimiento propuesto

3.1. Introducción

Para la validación y aceptación de la propuesta de procedimiento general para las pruebas de liberación de los productos de los centros de la facultad 1, basado en las especificaciones de NC ISO/IEC 9126-1:2005 presentadas en el capítulo 2, se utilizó el criterio de un grupo de especialistas de la Universidad de las Ciencias Informática. El panel se conformó con especialistas que poseen conocimientos en pruebas de calidad, muy hábiles en esta actividad y que tienen experiencias con proyectos que han sido liberados satisfactoriamente.

En el presente capítulo se hará la descripción de los pasos utilizados en la selección del panel de especialistas y los resultados obtenidos. Para el proceso de evaluación de la propuesta se tuvo en cuenta el proceso de selección de especialistas, la elaboración de la encuesta a aplicar y por último, los resultados de la evaluación. La calidad de los resultados depende, sobre todo, del cuidado que se ponga en la elaboración del cuestionario y en la elección de los especialistas consultados.

3.2. Proceso de selección de especialistas

Se entiende por especialista a una persona que conoce algo o domina un tema por la práctica, que posee experiencia en un campo y que es muy hábil en una actividad dada. Un especialista en una materia es capaz de interpretar correctamente las informaciones sobre dicho campo, de ofrecer valoraciones conclusivas de un problema en cuestión y de hacer recomendaciones al respecto. En el desarrollo de este proceso se consideraron tres etapas cruciales:

Determinar la cantidad de especialistas.

Conformar el listado de los especialistas.

Confirmar la participación de los especialistas.

Se tiene en cuenta que ningún especialista conoce las identidades y respuestas de los otros que componen el grupo, para lograr así que cada uno defienda sus opiniones y en caso de ser erróneas, no habrá pérdida de su prestigio.

3.2.1. Determinar la cantidad de especialistas y conformar el listado de especialistas

Se seleccionó una muestra de siete especialistas para la confección del panel, teniendo en cuenta el nivel de complejidad del contenido.

La confección del listado de especialistas se realizó atendiendo a la posibilidad real de participación de los candidatos, pues todos son profesionales de la UCI que tienen experiencia en la docencia, en el tema de la calidad de los productos de software y en el proceso productivo de la universidad. Poseen además, amplios conocimientos en temas relacionados con el proceso a evaluar:

Pruebas de liberación del software.

Calidad de software.

Existe una serie de cualidades propias de estos especialistas seleccionados, que se tuvieron en cuenta por parte de los autores de la investigación para la confección del listado. A continuación se relacionan:

Seriedad.

Honestidad.

Sinceridad.

Responsabilidad.

Creatividad.

Capacidad de análisis.

Estas cualidades han permitido que las opiniones brindadas sean confiables y válidas para el objetivo propuesto.

3.2.2. Confirmar la participación de especialistas

Una vez conformado el listado, se invitó personalmente a cada especialista elegido para participar en la evaluación a participar en un encuentro informal con el propósito de explicar en que consistía el trabajo en general, la propuesta a evaluar y el objetivo de la realización de la encuesta, así como el plazo de entrega. Una vez recibida la respuesta positiva, se estableció el listado final de los especialistas, informando a cada especialista su inclusión en el proceso a evaluar y las instrucciones necesarias para contestar las preguntas. De esta forma, culminó el proceso de selección, logrando la participación de los siete especialistas escogidos.

3.3 Elaboración de la encuesta

Las encuestas se llevaron a cabo de una manera anónima. Para la elaboración de la encuesta se tuvieron en cuenta los objetivos que debería cumplir la propuesta del procedimiento. Se les facilitó la posibilidad de modificar aspectos que ellos consideraran necesarios cambiar y presentar su opinión, a favor o en contra del procedimiento propuesto, con la libertad de expresar todo lo que se pudo obviar en la encuesta. (Anexo 13). Los autores del trabajo de la investigación, fueron explicando detalladamente a cada especialista, los objetivos y posibles resultados de la propuesta. Posteriormente se les entregó la encuesta con un plazo de tiempo para responderla. La encuesta establece una serie de preguntas que permiten visualizar la posibilidad real de aplicar la propuesta en los centros de la facultad 1.

3.4. Resultados de la evaluación

Una vez analizadas las siete encuestas realizadas a los especialistas, se obtuvieron los siguientes resultados:

El 100% de los especialistas considera que el procedimiento propuesto está a la altura de las necesidades de los centros de la facultad 1. Los especialistas 2: graduado de Ingeniería Informática en la UCI. Cuenta con 4 años de experiencia en temas de calidad y Gestión de software, 3: máster ,4: máster, 6: graduada de Ingeniera Informática en la UCI. Cuenta con 5 años de experiencia en temas de calidad. y 7: máster, argumentan que la propuesta de solución facilita el incremento de las pruebas de calidad, que representa una vía para tomar conciencia de la importancia de la calidad del software, basada en modelos de calidad.

Capítulo III

El 100% de los especialistas consideran que el desarrollo de las pruebas propuestas es lo suficientemente factible a las necesidades de los centros de la facultad 1. El especialista 3: graduada de Ingeniería Informática. Máster en Gestión de Proyectos. Cuenta con 7 años de experiencia en temas de calidad y Gestión de software, considera que el procedimiento mitiga si se usa correctamente los riesgos de requisitos mal definidos.

En la última pregunta aplicada a los especialistas, se evalúa una serie de criterios, dándoles una categoría de 1 a 5, donde uno representa el valor mínimo y 5 el máximo. La tabla 3.1 muestra los resultados de la misma.

Tabla 3.1. Resultados de las opiniones de los especialistas

| Criterio/Especialista | Especialista 1 | Especialista 2 | Especialista 3 | Especialista 4 | Especialista 5 | Especialista 6 | Especialista 7 |
|--|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Satisfacción a las necesidades de los centros de la facultad 1 | 5 | 5 | 5 | 4 | 5 | 5 | 5 |
| Adaptabilidad de la NC ISO/IEC 9126-1:2005 a los proyectos de los centros de la facultad 1. | 4 | 5 | 5 | 4 | 5 | 4 | 5 |
| Eficiencia de las herramientas seleccionadas para el desempeño de las pruebas. | 4 | 3 | 4 | 5 | 5 | 5 | 3 |
| Posibilidad de satisfacción de | 5 | 4 | 3 | 4 | 5 | 5 | 5 |

| | | | | | | |
|---|---|---|---|---|---|-----|
| <i>las actividades descritas en el procedimiento.</i> | | | | | | |
| <i>Posibilidad de implantación del procedimiento.</i> | 5 | 4 | 5 | 5 | 5 | 4 5 |

3.5. Resultados de las encuestas realizadas a los especialistas



Figura 3.1: Satisfacción a las necesidades de los centros de la facultad 1.

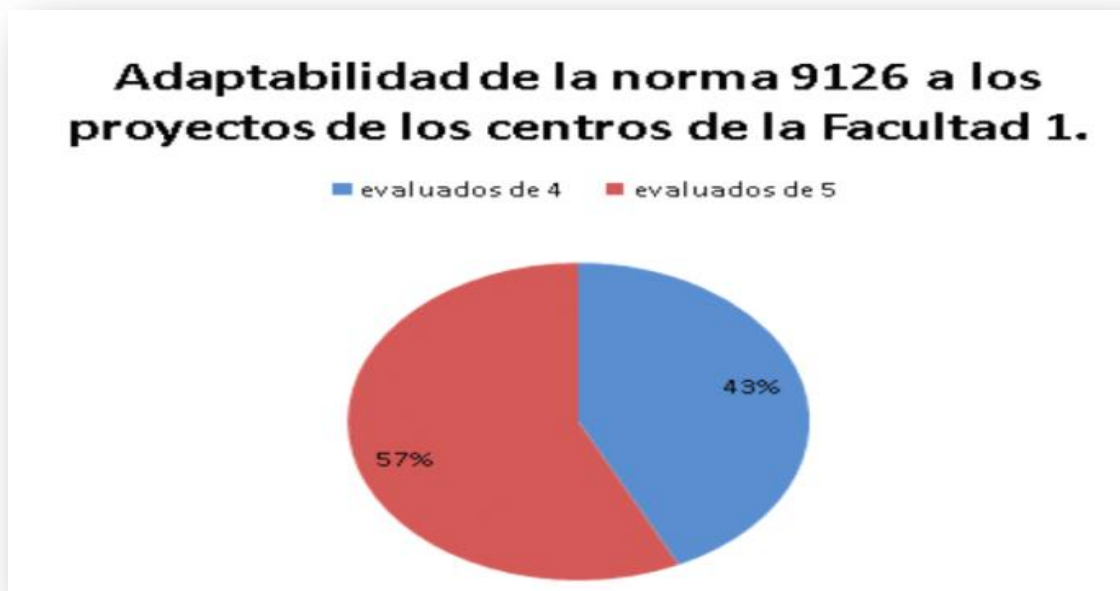


Figura 3.2: Adaptabilidad de la norma 9126 a los proyectos de los centros de la facultad 1.



Figura 3.3: Aplicación de la ficha con el resultado de las características.



Figura 3.4: Posibilidad de satisfacción de las actividades descritas en el procedimiento.

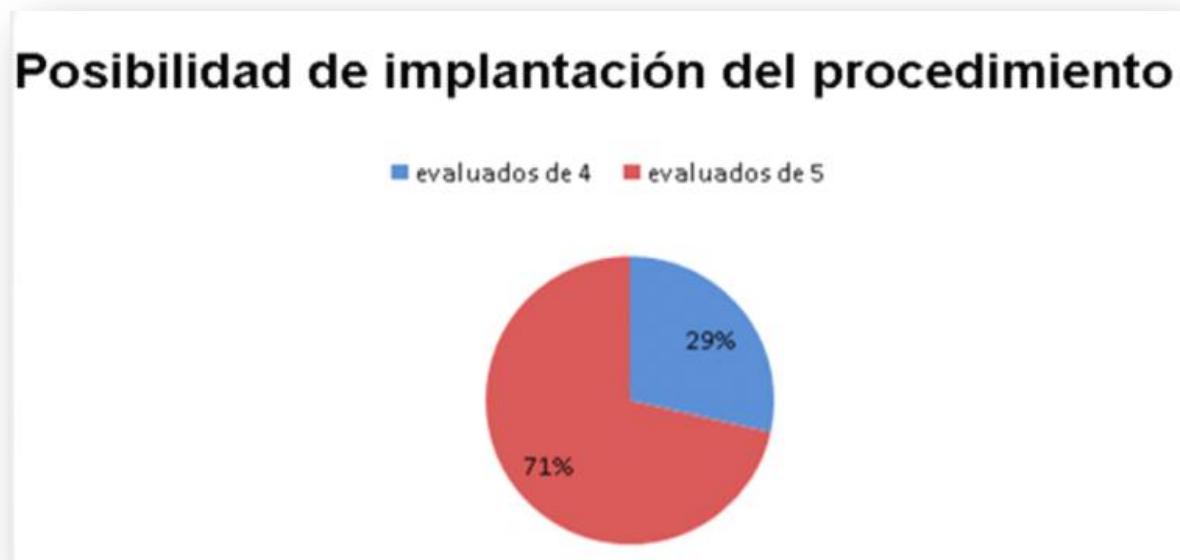


Figura 3.5: Posibilidad de implantación del procedimiento.

3.6. Beneficios de la propuesta

A partir del análisis realizado en los proyectos de los centros de la facultad 1 con vistas a solucionar la situación existente, y de los resultados arrojados de la validación realizada, se determina que la propuesta traerá los siguientes beneficios a los centros:

1. Normalización del proceso de pruebas de liberación para los productos de los centros de la facultad 1.
2. Mejora en la organización de las pruebas y en el aseguramiento de la calidad.
3. Aumento de la calidad de los productos y servicios.
4. Mitigación de los riesgos de requisitos mal definidos.
5. Documentación del proceso de evaluación de los mismos.

3.7. Conclusiones Parciales

Como parte de este capítulo se realizó la validación y aceptación de la propuesta de procedimiento general para las pruebas de liberación de los productos de la facultad 1, basado en las especificaciones de NC ISO/IEC 9126-1:2005 presentadas en el capítulo 2 mediante la técnica de “Panel de Expertos”. Se realizó una descripción de los pasos utilizados en la selección de los especialistas a participar en la validación, donde la selección estuvo encaminada a contar con los especialistas que tienen pleno conocimiento y experiencia en el área de las pruebas de calidad. Por último se graficaron los resultados obtenidos para un mejor análisis.

Conclusiones

Con el desarrollo de esta investigación se logró realizar un análisis documental sobre los métodos y técnicas empleados en Cuba y el mundo para las pruebas de software, conceptualizando términos indispensables en el campo de acción.

se logró profundizar en el estudio de las normas enfocadas a la evaluación de la conformidad de los productos de software, logrando definir como norma a utilizar la 9126:2005, por estar enfocada al producto de software y describir un modelo de calidad en dos partes, permite además evaluar los pasos para la liberación de los productos.

Se definió un procedimiento para el diseño, implementación y control de las pruebas de liberación de los productos de software de la facultad 1, permitiendo a los centros de producción de software de la facultad que los productos arriben al laboratorio industrial de prueba con la menor cantidad de defectos posibles.

Se logró Validar la propuesta del procedimiento utilizando la técnica de panel de experto específicamente el método de las encuestas.

Recomendaciones

- ✓ Implantar en los proyectos de los centros de la facultad 1, el procedimiento propuesto.
- ✓ Analizar otras herramientas con vistas a aplicar a las características de la NC ISO/IEC 9126-1:2005.
- ✓ Documentar los resultados de las pruebas de liberación, con vistas a tener un histórico para futuras pruebas.
- ✓ Aplicar la mejora continua al procedimiento.

Referencias bibliográficas

19011:2002, ISO. *Directrices para la auditoría de los sistemas de gestión de la calidad y/o ambiental.*

9126. NC ISO/IE 9126-1:2005. SISTEMAS DE GESTIÓN DE LA CALIDAD — REQUISITOS. 2005.

2000, ISO 9000:. *Sistemas de gestión de la calidad — Fundamentos y vocabulario.*

AENOR. 2008. [En línea] 19 de 11 de 2008. [Citado el: 05 de 11 de 2009.]

<http://www.infocalidad.net/noticias/noticias.asp?id=4867>.

Añasco, Ing. Juan Carlos. 2007. Pilar.com.ar. [En línea] 2007. [Citado el: 2009 de 12 de 02.]

<http://www.pilar.com.ar/industrias/temasgenerales/normas.htm>.

Antonio, Hidalgo Nuchera. 2004. Una introducción a la gestión de riesgos tecnológicos.

Gestión de la Innovación y de la Tecnología. Tribuna de debate. [En línea] 2004. [Citado el: 20 de 02 de 2009.] <http://www.madrimasd.org/revista/revista23/tribuna/tribuna1.asp>.

Bachtold, Ing. Mario Echandi. 2009. Global Standard Centroamerica. [En línea] 18 de 02 de

2009. [Citado el: 02 de 02 de 2009.] <http://www.globalstandardca.com/blog/>.

calidad, guía de la. Guiadelacalidad. [En línea] [Citado el: 03 de 03 de 2009.]

<http://www.guiadelacalidad.com/modelo-efqm.php>.

Cotton, Dr. Jesús Raúl Rodríguez. *Curso Herramientas de Mejoras.*

Crosby, Philip B. 2003 Reflexiones sobre calidad: 2003. MC Graw-hill. México

Cruz, Edel Avila. 2008. *Propuesta de procedimiento general para normalizar las pruebas en el desarrollo de Aplicaciones Web en la Facultad 7.* s.l. : UCI, 2008.

DCSSI, Documento editado por la oficina de consultoría de la DCSSI. 2003. *El software EBIOS.*

10 ATRACTIVOS PARA SEDUCIR. Paris : s.n., 2003.

de la Villa, Manuel, Ruiz, Mercedes y Ramos, Isabel. Modelos de Evaluación y Mejora de Procesos: Análisis Comparativo.

2007. Exelencia Empresarial. [En línea] 2007. [Citado el: 05 de 03 de 2009.]

http://web.jet.es/amoarrain/gestion_indicadores.htm.

guiadelacalidad. Guiadelacalidad. [En línea] [Citado el: 03 de 03 de 2009.]

<http://www.guiadelacalidad.com/modelo-efqm.php>.

ISO. 2005. ISO-9000-2005 Fundamentos y Vocabulario. 2005.

—. **2008.** NC ISO 9001:2008. SISTEMAS DE GESTIÓN DE LA CALIDAD — REQUISITOS. 2008.

—. **2000.** Selección y uso de la tercera edición de las normas ISO 9000. ISO. 2000.

Referencias Bibliográficas

- ISO, Secretaría del. 2008.** *Orientación para la Implementación de la.* 2008.
- ISO-9000-2005.** ISO-9000-2005 Fundamentos y Vocabulario. [aut. libro] ISO.
- 2008.** Kioskea.net. [En línea] 2008. [Citado el: 25 de 01 de 2009.]
<http://es.kioskea.net/contents/qualite/indicateurs.php3>.
- Lewis. 2005.** *Software Testing and Continuous Quality Improvement, AUERBACH.* 2005.
- Mainzer.** *Calidad y Formación.*
- Manuel de la Villa, Mercedes Ruiz, Isabel Ramos.** *Modelo de Evaluación y Mejoras de procesos. Análisis Comparativo.*
- Manuel García, Carlos Quispe. Luis Ráez. 2003.** *Mejora Continua de la Calidad en los Procesos.* Perú : s.n., 2003.
- Mejías, Carlos Alberto. 1998.** Planning. [En línea] Octubre de 1998. [Citado el: 03 de 03 de 2009.] <http://www.planning.com.co/bd/archivos/Octubre1998.pdf>.
- Pereiro, Jorge. 2008.** Portalcalidad. [En línea] 10 de 03 de 2008. [Citado el: 20 de 04 de 2009.]
<http://www.portalcalidad.com/modules/news/article.php>.
- Pillou, Jean-François. 2004.** Kioskea. [En línea] 16 de diciembre de 2004. [Citado el: 18 de 02 de 2009.] <http://es.kioskea.net/contents/qualite/iso-9001.php3>.
- Pressman, Roger. 1998.** "Ingeniería de Software. Un enfoque práctico". *"Ingeniería de Software. Un enfoque práctico"*. 1998.
- . *Ingeniería del Software. Un enfoque práctico.* Quinta Edición.
- Prof. Dr. Ing. Arturo Luis Romero, Lic. Sandor Luis Miranda. 2007.** LA CALIDAD, SU EVOLUCIÓN HISTÓRICA Y ALGUNOS CONCEPTOS Y TÉRMINOS ASOCIADOS . [En línea] 10 de 08 de 2007.
- Protocolo, Revista. 2006.** Protocolo. [En línea] 09 de 02 de 2006. [Citado el: 05 de 03 de 2009.] http://www.protocolo.com.mx/articulos.php?id_sec=4&id_art=465.
- RATIONAL. 2003.** *RATIONAL.* 2003.
- Rational. 2003.** *Rational Unifices Proccess.* 2003.
- Roberth G. Figueroa, camilo J. Solís, Armando A. Cabrera.** Metodologías tradicionales vs Metodologías ágiles .
- Roberth G. Figueroa, Camilo J. Solís, Armando A. Cabrera.** *Metodologías tradicionales vs metodologías ágiles.* s.l. : Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación.

Referencias Bibliográficas

Tejeda, Ing. Victor H. Rodríguez. 2007. *MEJORA CONTINUA PARA LA SATISFACCIÓN DEL CLIENTE.* 2007.

UNITEC.2003 8 principios de la Gestión Administrativa. [aut. libro] UNITEC- Dirección Estratégica y Proyectos.

Vásquez, Ing. Gerardo Rojas. 2000. ¿ CUÁN TO CONOCEMOS DE LAS NORMAS ISO 9000? 2000.

Vejerano, Juniedi García. 2007. Procedimiento para el piloto de una Solución Informática desarrollada en softel. Habana : s.n., 2007.

—. **2007.** *Procedimiento para el piloto de una Solución Informática en Softel.* s.l. : UCI, 2007.

Watson, Mary. 12 Manage the executive fast track. [En línea] [Citado el: 03 de 03 de 2009.] http://www.12manage.com/methods_demingcycle_es.html.

Bibliografía

9126. NC ISO/IE 9126-1:2005. SISTEMAS DE GESTIÓN DE LA CALIDAD — REQUISITOS. 2005.

AENOR. 2008. [En línea] 19 de 11 de 2008. [Citado el: 06 de 12 de 2009.]
<http://www.infocalidad.net/noticias/noticias.asp?id=4867>.

Añasco, Ing. Juan Carlos. 2007. Pilar.com.ar. [En línea] 2007. [Citado el: 05 de 12 de 2009.]
<http://www.pilar.com.ar/industrias/temasgenerales/normas.htm>.

Argentina, Seminario. IRTA. [En línea] [Citado el: 05 de 03 de 2010.]
http://www.irta.es/CAT/agenda/seminari_argentina_uruguay/comparacion_EFQM_ISO.pdf.

Bachtold, Ing. Mario Echandi. 2009. Global Standard Centroamérica. [En línea] 18 de 02 de 2009.
[Citado el: 18 de 02 de 2010.] <http://www.globalstandardca.com/blog/>.

Baquia. 2003. Baquia. [En línea] 28 de 10 de 2003. [Citado el: 04 de 03 de 2010.]
<http://www.baquia.com/com/20031028/bre00006.html>.

Brey, Gustavo A. 2006. IBM. [En línea] 2006. [Citado el: 20 de 01 de 2010.]
<http://www.ibm.com/es/>.

calidad, guía de la. Guiadelacalidad. [En línea] [Citado el: 06 de 12 de 2009.]
<http://www.guiadelacalidad.com/modelo-efqm.php>.

Cotton, Dr. Jesús Raúl Rodríguez. *Curso Herramientas de Mejoras.*

Cruz, Edel Avila. 2008. *Propuesta de procedimiento general para normalizar las pruebas en el desarrollo de Aplicaciones Web en la Facultad 7.* s.l. : UCI, 2008.

Crosby, Philip B. 2003 Reflexiones sobre calidad: 2003. MC Graw-hill. México

de la Villa, Manuel, Ruiz, Mercedes y Ramos, Isabel. Modelos de Evaluación y Mejora de Procesos: Análisis Comparativo.

Empresario, Programa de Calidad Total-Cuaderno de Herramientas-Guía del. Caja de Herramientas. La lista de verificación. [En línea] http://www.infomipyme.com/Docs/GENERAL/Offline/GDE_07.htm.

ENARCO. Herramientas de Control de Calidad y Mejora Continua. [En línea] [Citado el: 20 de 02 de 2010.] <http://www.mitecnologico.com/Main/HerramientasDeControlDeCalidadYMejoraContinua>.

2007. Exelencia Empresarial. [En línea] 2007. [Citado el: 05 de 03 de 2010.]
http://web.jet.es/amozarrain/gestion_indicadores.htm.

guiadelacalidad. Guiadelacalidad. [En línea] [Citado el: 03 de 11 de 2009.]
<http://www.guiadelacalidad.com/modelo-efqm.php>.

Informativos, Compendios. Compendios informativos. Gestión por procesos. Mejora e Innovación de Procesos. *Compendios informativos. Gestión por procesos. Mejora e Innovación de Procesos.* [En línea] http://www.delfos.co.cu/boletines/bsa/content_como_inform%20gest%20proc1.html.

IRCA. www.irca.org. *Guías del Grupo de Práctica de Auditorías de ISO/ IAF.* [En línea] [Citado el: 14 de 11 de 2009.] <http://www.irca.org/downloads/IRCA250d%20APG%20ChecklistSP.pdf>.

ISO. 2005. ISO-9000-2005 Fundamentos y Vocabulario. 2005.

—. **2008.** NC ISO 9001:2008. Sistemas de Gestión de la Calidad — Requisitos. 2008.

—. **2000.** Selección y uso de la tercera edición de las normas ISO 9000. *ISO.* 2000.

ISO, Secretaría del. 2008. *Orientación para la Implementación de la norma ISO.* 2008.

ISO-9000:2005. ISO-9000:2005 Fundamentos y Vocabulario. [aut. libro] ISO.

KAIZEN, Grupo. Herramientas para la Mejora Continua. *Herramientas para la Mejora Continua.* [En línea] [Citado el: 18 de 1 de 2010.] <http://www.gestiopolis.com/canales5/ger/gksa/docs/2.pdf>.

2008. Kioskea.net. [En línea] 2008. [Citado el: 25 de 01 de 2010.] <http://es.kioskea.net/contents/qualite/indicateurs.php3>.

Lewis. 2005. *Software Testing and Continuous Quality Improvement, AUERBACH.* 2005.

Mainzer. *Calidad y Formación.*

Manuel de la Villa, Mercedes Ruiz, Isabel Ramos. *Modelo de Evaluación y Mejoras de procesos. Análisis Comparativo.*

Manuel García, Carlos Quispe. Luis Ráez. 2003. *Mejora Continua de la Calidad en los Procesos.* Perú : s.n., 2003.

Pereiro, Jorge. 2008. Portalcalidad. [En línea] 10 de 03 de 2008. [Citado el: 15 de 12 de 2009.] <http://www.portalcalidad.com/modules/news/article.php>.

Pillou, Jean-François. 2004. Kioskea. [En línea] 16 de diciembre de 2004. [Citado el: 18 de 02 de 2010.] <http://es.kioskea.net/contents/qualite/iso-9001.php3..>

Pressman, Roger. 1998. "Ingeniería de Software. Un enfoque práctico". *"Ingeniería de Software. Un enfoque práctico"*. 1998.

—. *Ingeniería del Software. Un enfoque práctico.* Quinta Edición.

Prof. Dr. Ing. Arturo Luis Romero, Lic. Sandor Luis Miranda. 2007. La Calidad, su evolución histórica y algunos conceptos y términos asociados. [En línea] 10 de 08 de 2007.

Protocolo, Revista. 2006. Protocolo. [En línea] 09 de 02 de 2006. [Citado el: 05 de 03 de 2010.] http://www.protocolo.com.mx/articulos.php?id_sec=4&id_art=465.

RATIONAL. 2003. *RATIONAL.* 2003.

Rational. 2003. *Rational Unifices Process.* 2003.

Registrars, Perry Johnson. 2007. PERRY JOHNSON REGISTRARS. *PJR.* [En línea] 2007. [Citado el: 02 de 02 de 2010.] <http://www.pjr.com/spanish/iso9001.htm>.

Roberth G. Figueroa, camilo J. Solís, Armando A. Cabrera. Metodologías tradicionales vs Metodologías ágiles .

Roberth G. Figueroa, Camilo J. Solís, Armando A. Cabrera. *Metodologías tradicionales vs metodologías ágiles.* s.l. : Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación.

Sagi, Diego J. Bodas. Ingeniería Técnica Informática de Sistemas. CES Felipe II (UCM). [En línea] <http://www.computing-es.com>.

SEDIC. Gestión de calidad y mejora de procesos. Aplicación a bibliotecas y centros de documentación. *Herramientas de mejora continua. Diagrama de Procesos.* [En línea] <http://www.sedic.es/autoformacion/seccion6DProcesos.htm>.

Tejeda, Ing. Víctor H. Rodríguez. 2007. *Mejora Continua para la satisfacción del cliente.* 2007.

UNITEC.2003 8 principios de la Gestión Administrativa. [aut. libro] UNITEC- Dirección Estratégica Proyectos.

Vásquez, Ing. Gerardo Rojas. 2000. ¿Cuánto conocemos de las normas ISO 9000? 2000.

Watson, Mary. 12 Manage the executive fast track. [En línea] [Citado el: 03 de 03 de 2010.] http://www.12manage.com/methods_demingcycle_es.html.

Glosario de Términos

Aplicación Web: Se puede definir como una Aplicación en la cual un usuario por medio de un navegador realiza peticiones a una aplicación remota accesible a través de Internet (o a través de una intranet) y que recibe una respuesta que se muestra en el propio navegador.

Producto software: Está formado por el software y los manuales de usuario y de entrenamiento.

Prueba: Actividad en la cual un sistema o uno de sus componentes se ejecutan en circunstancias previamente especificadas, los resultados se observan y registran y se realiza una evaluación de algún aspecto.

Procedimiento: Forma especificada para llevar a cabo una actividad o un proceso.

No Conformidad: Incumplimiento de un requisito.

Evaluación: Es el medio fundamental para conocer la relevancia social de los objetivos planteados, el grado de avance con respecto a los mismos, así como la eficacia, impacto y eficiencia de las acciones realizadas.

ISO (International Organization for Standardization): Es una organización internacional no gubernamental, compuesta por representantes de los organismos de normalización nacionales, que produce normas internacionales industriales y comerciales. Dichas normas se conocen como **normas ISO** y su finalidad es la coordinación de las normas nacionales, en consonancia con el Acta Final de la Organización Mundial del Comercio, con el propósito de facilitar el comercio, facilitar el intercambio de información y contribuir con unos estándares comunes para el desarrollo y transferencia de tecnologías.

Metodología: Es la parte del proceso de investigación que sigue a la propedéutica y permite sistematizar los métodos y las técnicas necesarios para llevarla a cabo. Es pues, una etapa, una parte del proceso.

Modelo de Calidad: El modelo de calidad consiste en reunir todas las actividades y funciones en forma tal que ninguna de ellas esté subordinada a las otras y que cada una se planee, controle y ejecute de un modo formal y sistemático.

Rol: Papel, cometido o función que tiene o desempeña que interpreta un actor.

Feedback (retroalimentación): Es un proceso por el que una cierta proporción de la señal de salida de un sistema se redirige de nuevo a la entrada. Los ejemplos de la realimentación se pueden encontrar en la mayoría de los sistemas complejos, tales como ingeniería, arquitectura, economía, y biología.

Firmware o programación en firme: Es un bloque de instrucciones de programa para propósitos específicos, grabado en una memoria de tipo no volátil.

Software: Se refiere al equipamiento lógico o soporte lógico de una computadora digital, y comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de tareas específicas; en contraposición a los componentes físicos del sistema, llamados hardware.

Base de Datos: Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Norma: Una norma o estándar es una especificación que reglamenta procesos y productos para garantizar la interoperabilidad. Más específicamente, una norma de calidad es una regla o directriz para las actividades, diseñada con el fin de conseguir un grado óptimo de orden en el contexto de la calidad.

Framework: En el desarrollo de software, un framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado.

Documentación del Producto: Documentos necesarios para que el producto sea evaluado, como son: Manual de Usuario, Manual de Instalación, Expediente del Producto, Instalación del Producto de Software.

Informe final: Documento que presenta resultados obtenidos o proporciona evidencia de actividades desempeñadas.

Modelo de calidad del software: Permite uniformar la filosofía de trabajo, para lograr una mayor confiabilidad, mantenibilidad 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.

Normas y Estándares: Son acuerdos documentados que contienen especificaciones técnicas u otros criterios precisos para ser usados consistentemente como reglas,

guías, o definiciones de características, para asegurar que los materiales, productos, procesos y servicios se ajusten a su propósito.