



Universidad de las Ciencias
Informáticas

Universidad de las Ciencias Informáticas

Facultad 4

*Título: Proceso de evaluación de la documentación para el servicio de Pruebas
Funcionales a nivel de Sistema que ofrece el Grupo de Calidad de
FORTEQ.*

Autora: Arianna Rodríguez Jiménez

Tutora: Ing. Greisy Gálvez George

La Habana, Cuba

Junio 2011

"Año 53 de la Revolución"

DECLARACIÓN DE AUTORÍA

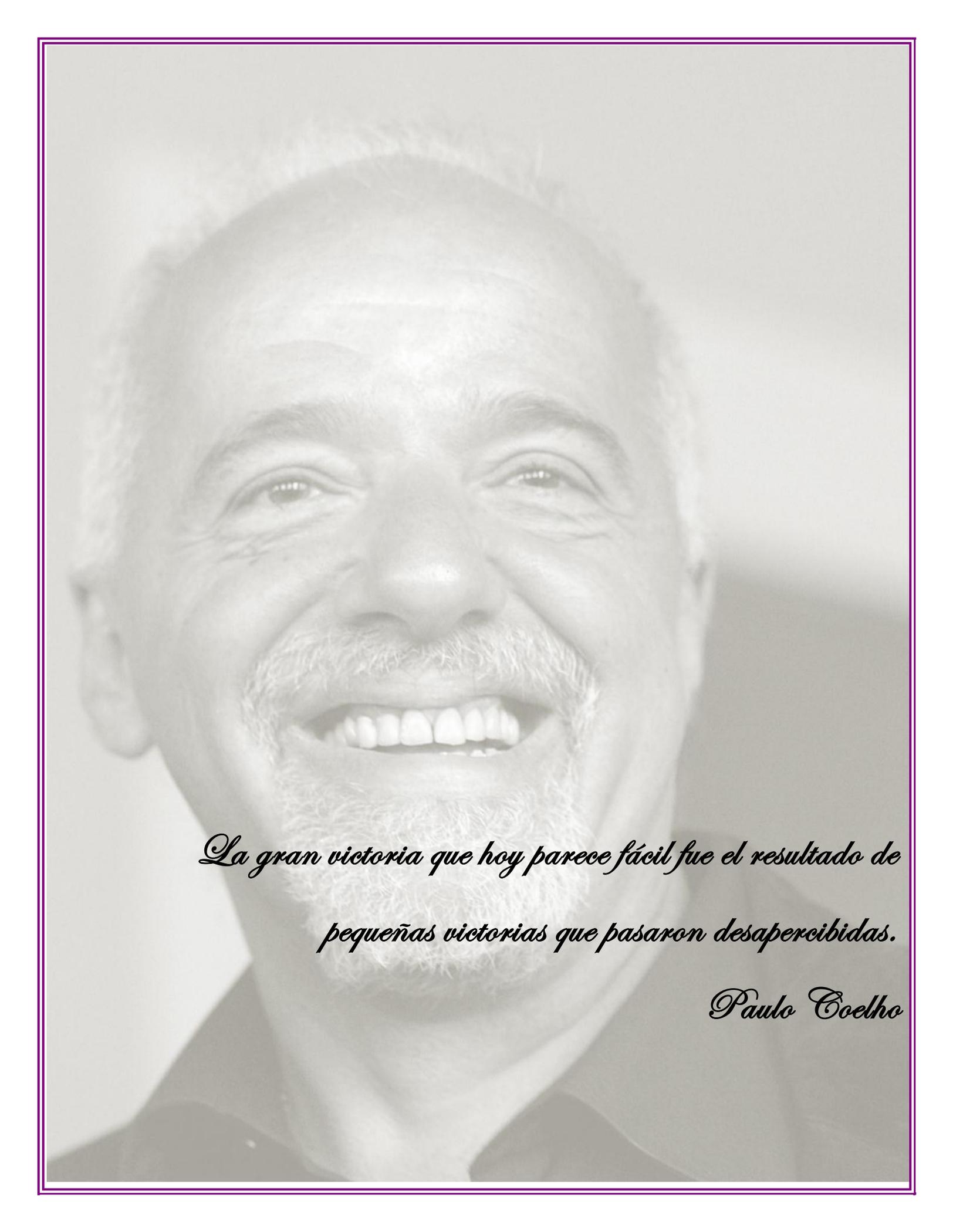
Por este medio declaro que soy la única autora del trabajo de diploma: Proceso de evaluación de la documentación para el servicio de Pruebas Funcionales a nivel de Sistema que ofrece el Grupo de Calidad de FORTES y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio. Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Autora: _____

Arianna Rodríguez Jiménez

Tutora: _____

Greisy Gálvez George



*La gran victoria que hoy parece fácil fue el resultado de
pequeñas victorias que pasaron desapercibidas.*

Paulo Coelho

Dedicatoria

A mi mamá y mis abuelos, por el esfuerzo, sacrificio, confianza y apoyo a lo largo de estos años.

A mi hermano, para quien aspiro ser un modelo a seguir

De forma muy especial a mi papá, por ser la luz de mi vida, mi ejemplo a seguir en todo momento y la fuerza que me alienta a levantarme cada mañana tratando de ser mejor persona cada día.

Agradecimientos

Primeramente agradezco a la Revolución, y de forma especial al compañero Fidel Castro, por darme la oportunidad de estudiar en una universidad tan maravillosa como esta. A la facultad 8, por permitirme hacer realidad mis sueños.

No me alcanzarían las palabras para decirle a mi mamá cuánto le agradezco la infinidad de sacrificios, esfuerzos, la constancia, de dedicación, apoyo y amor incondicional que me ha dado desde mi primer instante de vida.

A mi papá por su confianza, por no permitir que me rindiera nunca, por ser incansable en su lucha para hacer de mí una persona mejor, por ser la persona que más admiro en el mundo, a la que más quiero y de quien más orgullosa me siento.

A mi abuela María por ser mi cómplice, amiga, compañera, mi otra mamá, por ser mi refugio y consuelo, como suelo decirle: mi otra mitad que anda por el mundo, amén de que mi papá se ponga celoso. Por apoyarme y malcriarme tanto con todas mis ideas locas y mis sueños, por no dejar de creer en mí nunca, aun cuando ni yo misma me creía capaz de superar los obstáculos.

Al mi abuelo, mi tío, mis abuelos paternos y mi Nana, el amor, la confianza y la dedicación.

Al resto de mi familia, en especial a los que creían que no podría lograrlo, muchas gracias, el deseo de demostrarles que se equivocaban me hizo mantenerme firme en momentos en los que quería darme por vencida, a los que creyeron en mí, gracias por todo.

A alguien que ha sido amigo, hermano y compañero en todas las cosas locas que he vivido a lo largo de estos 5 años, Yasmany Aguilera (El Flako), no tengo palabras... gracias por todo. A Yerisleydi Sao (Yeri) o mi manager como le gusta decir, por hacer de mi quinto año el más divertido y emocionante, por atravesar conmigo momentos difíciles y estar ahí cada vez que la necesito. Espero poder devolverles algún día al menos la parte más pequeña de todo lo que han hecho por mí.

A Javy, por todo lo bueno y malo que hemos pasado juntos, en especial por las cosas malas, me hicieron crecer, madurar y hacerme fuerte ante las dificultades. Gracias por hacer de mi vida algo maravilloso con tu presencia.

A las personas que han estado conmigo desde el principio, atravesando buenos y malos momentos, Lisbet Álvarez, Addiel Rodríguez, Tomás López, gracias por estar ahí cuando más los necesité, y a pesar

de mis majaderías, nunca darme la espalda. Gracias también a otros que ya no están aquí, Lisy, Vilma, Karel, espero verlos nuevamente y agradecerles personalmente.

Por último muchas gracias al profesor Yudislandry, por ser un padre para mí, por la dedicación, constancia, por nunca rendirse en su afán de hacernos hombres y mujeres del futuro, por inspirarnos respeto, cariño, comprensión y hacerse partícipe de todos nuestros problemas brindándonos su mano cuando la necesitábamos, aún cuando por pena y respeto no se lo decíamos. Espero que se sienta orgulloso de su grupo, de sus muchachos, como suele decir, por mi parte, haré lo imposible por no defraudar la confianza que me ha depositado.

A todas las personas que de una forma u otra han influido en la persona que soy hoy...Muchas gracias por todo.

ARY.

RESUMEN

La presente investigación centra como objeto de estudio los procesos de pruebas de software y dentro de estos el proceso de evaluación de pruebas de software. El objetivo general de la investigación es contribuir a que la documentación que sirve de entrada al proceso de pruebas funcionales a nivel de sistema que desarrolla el Grupo de Calidad de FORTES reúna los requisitos para la evaluación de los diferentes proyectos productivos del centro. Del estudio de varias metodologías de diseño de procesos se toma Six Sigma para generar una propuesta que involucre a las actividades de evaluación de la documentación. La aplicación de la propuesta de solución diseñada arrojó como resultado su efectividad, dándole solución a la problemática que se planteaba al inicio de la misma.

Palabras Claves: calidad, documentación, evaluación, metodologías de diseño, procesos, pruebas.

ÍNDICE

INTRODUCCIÓN..... 1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA..... 4

1.1 INTRODUCCIÓN4

1.2 CALIDAD DE SOFTWARE.....4

1.3 PRUEBAS DE SOFTWARE.....6

 1.3.1 *Herramientas documentadas*11

 1.3.2 *Pruebas Exploratorias*15

1.4 EVALUACIÓN DE LA CALIDAD.....16

 1.4.1 *Métricas de Software*18

1.5 PROCESOS DE PRUEBA DE SOFTWARE25

 1.5.1 *Metodologías para el diseño y mejora de procesos*.....31

1.6 CONCLUSIONES DEL CAPÍTULO35

CAPÍTULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN PROCESO DE EVALUACIÓN DE LA DOCUMENTACIÓN PARA EL SERVICIO DE PRUEBAS FUNCIONALES A NIVEL DE SISTEMA QUE OFRECE EL GRUPO DE CALIDAD DE FORTES..... 36

2.1 INTRODUCCIÓN36

2.2 DESCRIPCIÓN DEL PROCESO DE EVALUACIÓN37

 2.2.1 *Ejecución de Pruebas Exploratorias*.....39

2.3 DESCRIPCIÓN DE LAS ACTIVIDADES.....41

2.4 DESCRIPCIÓN DE LOS PASOS DE PRUEBAS A SEGUIR PARA LAS REVISIONES49

2.5 ORGANIZACIÓN DEL EQUIPO DE TRABAJO51

2.6 CONCLUSIONES DEL CAPÍTULO52

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN PROCESO DE EVALUACIÓN DE LA DOCUMENTACIÓN PARA EL SERVICIO DE PRUEBAS FUNCIONALES A NIVEL DE SISTEMA QUE OFRECE EL GRUPO DE CALIDAD DE FORTES..... 53

3.1 INTRODUCCIÓN53

3.2 APLICACIÓN DE LA SOLUCIÓN PROPUESTA.....53

 3.2.1 *Casos de estudio*.....53

3.3 RESULTADOS DE LA APLICACIÓN.....54

3.4 ANÁLISIS DE LOS CASOS DE ESTUDIO.....57

 3.4.1 *Análisis comparativo con la situación anterior*.....59

3.5 CONCLUSIONES DEL CAPÍTULO61

CONCLUSIONES GENERALES..... 62

RECOMENDACIONES..... 63

TRABAJOS CITADOS 64

BIBLIOGRAFÍA..... 65

ANEXOS..... 67

INTRODUCCIÓN

La actualidad productiva en el contexto actual del siglo XXI es inexistente sin el desarrollo de las tecnologías computacionales. La fabricación de software es por tanto el sustento del presente y el futuro de cualquier nación, en especial en países subdesarrollados, como Cuba. La elaboración de software se convierte en una actividad cada vez más demandada, lo que provoca que las empresas dedicadas a la producción de bienes y servicios informáticos tiendan a buscar las vías idóneas para el perfeccionamiento de la gestión empresarial.

Las principales sedes productoras de conocimientos y nuevas tecnologías son sin dudas las universidades. En Cuba existen varios centros de enseñanzas de nivel superior con una historia consolidada en este ámbito, pero es a partir de la fundación de la Universidad de las Ciencias Informáticas (UCI) en el año 2002 cuando la producción de software y otros productos informáticos convierten a la universidad cubana en un centro de producción de vital importancia para el desarrollo tecnológico y económico del país.

Siendo las tecnologías computacionales las dominantes en el contexto actual, la calidad del producto y la documentación que lo argumenta es un medio fundamental para introducir las producciones de la UCI en el mercado mundial actual. En la UCI actualmente se desarrollan proyectos de gran envergadura y con este fin surge el Centro de Tecnologías para la Formación (FORTES), el cual alberga una amplia gama de productos encargados de fortalecer los medios de enseñanza dentro y fuera de nuestro país.

Con la finalidad de elevar la calidad de los productos que se desarrollan, el centro cuenta con el Grupo de Calidad FORTES (GCF), cuya misión es alcanzar de forma eficaz y eficiente los siguientes objetivos:

- Ofrecer servicios de pruebas de software a todos los proyectos productivos de FORTES e interesados, realizando actividades de verificación en correspondencia con los roles definidos.
- Formar profesionales competentes, especializados en pruebas de software desde la práctica laboral en el área productiva de FORTES y con la Dirección del GCF. (1)

Bajo la premisa de darle cumplimiento a los objetivos anteriormente mencionados, se ha desarrollado un proceso de pruebas cuyo servicio fundamental es la realización de pruebas a nivel de sistema, el mismo está complementado por dos procesos estratégicos:

- Auditorías y Revisiones periódicas
- Evaluación preventiva de la documentación (1)

La implementación de estos procesos estratégicos se ha visto afectada por dificultades presentes en los documentos, tales como falta de datos imprescindibles para la comprensión de la información, identificando los probadores no conformidades que no se corresponden con la versión del producto en prueba, o se revisan aspectos que no se ajustan a la iteración que se está realizando; ocasionando todo esto mayor complejidad en el trabajo de los probadores durante la revisión del producto y tardanza por la espera del completamiento de los datos deficientes en la documentación inicialmente entregada. De la cantidad de no conformidades identificadas en el curso 2009-2010, aproximadamente el 60% estaban relacionadas con problemas en la documentación.

Por lo anteriormente expuesto se puede determinar el siguiente problema a resolver: ¿Cómo contribuir a que la documentación que sirve de entrada al proceso de pruebas funcionales a nivel de sistema que desarrolla el GCF reúna los requisitos para la evaluación de los diferentes proyectos productivos de FORTES? Para darle solución al problema se definen como objeto de estudio los Procesos de Pruebas de Software y como campo de acción el Proceso de evaluación de pruebas de software. El objetivo general es: Contribuir a que la documentación que sirve de entrada al proceso de pruebas funcionales a nivel de sistema que desarrolla el GCF reúna los requisitos para la evaluación de los diferentes proyectos productivos de FORTES, mediante la creación de un proceso para la evaluación de los mismos. Como idea a defender se plantea que: Aplicando un proceso de evaluación de la documentación que sirve como entrada a los proyectos productivos en el proceso de pruebas funcionales a nivel de sistema que implementa el GCF entonces se podría alcanzar el cumplimiento de los requisitos necesarios en la documentación establecida.

Para darle cumplimiento al objetivo general se definen las siguientes tareas de investigación:

- Selección de la bibliografía necesaria para abordar temas relacionados con los procesos que existen definidos en el mundo y en la UCI para evaluar la documentación en los proyectos de desarrollo de software.
- Realización de un estudio sobre la documentación generada por los proyectos que tributa al proceso de pruebas.
- Desarrollo de un proceso para la evaluación de la documentación que atañe a las pruebas de los proyectos de desarrollo en el centro FORTES.
- Aplicación del proceso desarrollado para la evaluación de la documentación que sirve como entrada al proceso de pruebas del GCF.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

Una de las grandes problemáticas en el desarrollo de software a nivel mundial es el control de la calidad. Cuando se habla de Calidad de Software es imprescindible remitirse a las pruebas de software como la vía óptima para la detección de errores en el ciclo de desarrollo del mismo (2).

Para garantizar que los probadores puedan revisar el software, es necesario contar con documentos que expliquen de forma detallada los aspectos que se deben conocer sobre el producto que está probando y cómo este debe ser, ya que de las diferencias entre lo que es y lo que debería ser, surgen las no conformidades. Por esta razón, la calidad con que se realizan estos documentos es de crucial importancia en el resultado final, por lo que los mismos también deben ser evaluados y revisados.

En este capítulo se expone un estudio detallado acerca de lo anteriormente expuesto. Se analizan temas como: calidad de software, pruebas de software, documentación relacionada con el proceso de pruebas entre otros.

1.2 Calidad de software

El Dr. Armand V. Feigenbaum (2), introduce en la década de los '50 el término de calidad total. La idea es de producir con calidad desde que inicia el proceso productivo, más que inspeccionar y controlar la calidad después de elaborado el producto.

Este concepto incluye los siguientes puntos:

- La calidad tiene que ser planeada completamente con base en un enfoque orientado hacia la excelencia en lugar del enfoque tradicional orientado hacia la falla.
- Todos los miembros de la organización son responsables de la calidad.

Compromiso de la organización.

- Sistema efectivo para integrar los esfuerzos del desarrollo, mantenimiento y mejoramiento de la calidad.
- Definición de estándares, evaluación del cumplimiento de los estándares, corrección cuando el estándar no se ha cumplido y plantación para mejorarlos.

Integración de actividades (2):

- Las mejoras de la calidad más importantes provienen de ideas del personal. Para el control de la calidad y el mejoramiento de procesos se deben utilizar herramientas estadísticas.
- La automatización no es la solución a los problemas de calidad. Las actividades humanas son fundamentales en cualquier programa de calidad total.
- Cada integrante de la organización tiene que poder controlar su propio proceso y ser completamente responsable de calidad.

Para que el control de calidad sea efectivo, debe iniciarse con el diseño del producto y terminar sólo cuando se encuentre en manos de un consumidor satisfecho. (2)

Estos principios son contemplados en las actividades que agrupa el aseguramiento o garantía de la calidad el cual consiste en la auditoría y las funciones de información de la gestión, su objetivo es proporcionar la gestión para informar de los datos necesarios sobre la calidad del producto.

Según Pressman (3), la calidad del software es la concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente. El estándar 610 de la IEEE plantea que la calidad de software es el grado con que un sistema o componente cumple con los requerimientos especificados y las necesidades o expectativas del cliente o usuario (4).

Carlos Pes (5) en su minidiccionario informático plantea que la calidad de un programa se puede medir en base a tres aspectos principales:

1. Sus características operativas: Se debe valorar si el software hace lo que se espera de él (corrección) y si, para ello, se utilizan, óptimamente, los recursos de la computadora (eficiencia), tales como: la memoria, el tiempo de CPU, etc. También se debe evaluar si la aplicación ofrece una interfaz adecuada al usuario (facilidad de uso) y si es seguro con respecto a los datos (integridad).

2. Su capacidad para sufrir cambios: En este sentido, es importante estimar en qué medida el programa es susceptible de ser corregido (facilidad de mantenimiento) o cambiado (flexibilidad). También hay que ver si resulta fácil hacer pruebas de su funcionamiento (facilidad de prueba).
3. Su adaptabilidad a entornos distintos: Hay que preguntarse hasta qué punto se podría volver a usar parte de dicho software en otro proyecto (reusabilidad). Asimismo, se debe valorar si el software puede interactuar con otros sistemas informáticos (facilidad de interoperación) y si se puede usar en otra máquina que utilice un procesador distinto (portabilidad), aunque sea realizando pequeños cambios en el software.

Para el usuario, calidad es la aptitud para el uso, no la conformidad con las especificaciones. El usuario definitivo raramente conoce lo que está en las especificaciones. Su valoración de la calidad se basa en si el producto es apto para el uso cuando se lo entreguen a él, y si luego continúa siéndolo (6).

Luego del análisis de los conceptos anteriores, y en concordancia con lo expresado por Pressman y los estándares de la IEEE, se puede decir que la calidad está dada por el conjunto de características que le confieren su capacidad de satisfacer las expectativas del cliente, acorde al cumplimiento de una serie de requisitos establecidos previamente, haciendo hincapié en dos puntos importantes:

- Los requisitos del software son la base de las medidas de la calidad. La falta de concordancia con los requisitos es una falta de calidad.
- Los estándares definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la ingeniería del software. Si no se siguen esos criterios, casi siempre habrá falta de calidad.

La garantía o aseguramiento de la calidad de software (SQA), es un diseño planificado y sistemático de acciones que se requieren para asegurar la calidad del software. Entre las actividades de más peso dentro de este diseño se encuentran las pruebas de software.

1.3 Pruebas de Software

La incorporación del control de la calidad al proceso de desarrollo de software ha provocado un cambio sustancial en la forma de producir los mismos, puesto que representa un aumento de los beneficios

obtenidos por la realización del mismo. Permite que el cliente eleve su satisfacción con la fiabilidad del software, así como una disminución de errores durante su explotación.

El software es un producto eminentemente intelectual, que depende al 100% de la pericia y formación de las personas que lo desarrollan pero las personas cometen errores o cambian de idea, por lo que se hace necesario probar.

El proceso de pruebas de software es normalmente subestimado en importancia, magnitud y alcance al momento de planificar y ejecutar proyectos de software. Las consecuencias de esta subestimación son siempre vistas en la fase de despliegue del software cuando ocurren errores no detectados en la fase de construcción y pruebas, esto acarrea pérdidas económicas e incrementa la falta de credibilidad de las soluciones de tecnología de información.

El Diccionario de la Real Academia de la Lengua Española define la palabra prueba como un ensayo o experimento que se hace de algo para saber cómo resultará en su forma definitiva (7). Las pruebas consisten en realizar ensayos del funcionamiento de las aplicaciones en un ambiente controlado por el probador, a fin de detectar la mayor cantidad de errores posibles antes de que el producto llegue a manos de los usuarios finales, para evitar que cuando sea puesto en funcionamiento pueda generar cualquier tipo de fallo.

Las pruebas son la verificación del comportamiento de un sistema, basado en la ejecución de un conjunto de acciones controladas. Numerosos autores han hecho referencia a este proceso imprescindible para alcanzar la calidad del software que se produce.

Según Pressman (3) las pruebas del software son un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación. La IEEE las define como una actividad en la cual un sistema o componente es ejecutado bajo condiciones específicas, se observan o almacenan resultados y se realiza una evaluación de algún aspecto del sistema o componente (4).

Para lo que a esta investigación respecta y concordando plenamente con lo establecido por la IEEE, las pruebas son una actividad por la cual un sistema se ejecuta en circunstancias previamente especificadas, los resultados se observan y se registran y se realiza una evaluación de algún aspecto. Es importante aclarar que las pruebas suelen mostrar la presencia de errores, pero nunca demuestran su ausencia.

Existen diferentes niveles de pruebas, los cuales se realizan en diferentes momentos del ciclo de vida del software. Estos organizan e integran el conjunto de técnicas y pruebas a realizar (3):

- Pruebas Unitarias: Son pruebas a menor escala, son las encargadas de probar el correcto funcionamiento de un módulo de código.
- Pruebas de Integración: La prueba de integración se lleva a cabo durante la construcción del sistema, involucra a un número creciente de módulos y termina probando el sistema como conjunto.
- Pruebas de Sistema: Es el proceso de integrar el sistema y el hardware para verificar que el sistema reúna los requerimientos especificados. Una prioridad de este tipo de pruebas es concentrarse más en la capacidad del sistema, que en la capacidad de los componentes, es decir, en el uso de la interacción de las funciones que en probar los detalles de implementación; otra prioridad es probar las situaciones críticas en lugar de casos especiales.
- Pruebas de Aceptación: Son pruebas realizadas por el cliente. Son básicamente pruebas funcionales, sobre el sistema completo y buscan una cobertura de la especificación de requisitos y del manual de usuario.
- Pruebas de Regresión: Es un conjunto de pruebas que se ejecutan nuevamente. Su objetivo es verificar que cambios ocurridos en un componente del sistema no provocan errores adicionales en otros componentes del mismo que ya han sido probados.

A continuación se describen brevemente los diferentes tipos de pruebas agrupadas según su nivel correspondiente, haciendo énfasis en las pruebas funcionales a nivel de sistema, ya que son la actividad fundamental del GCF:

- Pruebas a nivel de unidad (3):

- Pruebas de interfaz: Estas pruebas analizan la interfaz del módulo correspondiente para probar si la información fluye correctamente hacia y desde la unidad de programa que está siendo probada.
- Prueba de estructura de datos: El análisis de las estructuras de datos locales tiene el propósito de comprobar si los datos que se mantienen temporalmente conservan su integridad durante todos los pasos de ejecución del algoritmo.
- Condiciones límites: Se prueban las condiciones límites para comprobar la correcta ejecución de los módulos en los límites establecidos de condiciones como de restricciones de procesamiento.
- Caminos independientes: Una vez que se ejercitan todos los caminos independientes o básicos de la estructura de control del módulo se asegura probar con su ejecución, al menos una vez, todas las sentencias del mismo.
- Pruebas a nivel de integración (3):
 - Integración descendente: Parte del módulo de control principal, para ir luego incorporando los módulos subordinados progresivamente.
 - Integración ascendente: Se integran y prueban los componentes de los niveles más bajos de la estructura del programa, antes del desarrollo de los niveles altos.
 - Prueba de regresión: Es aplicada después de la prueba de integración, en la medida que se añada un nuevo módulo.
 - Prueba de humo: Se aplica en productos de software desarrollados por paquetes, generalmente, proyectos críticos por tiempo. No es muy exhaustiva, pero permite descubrir errores importantes.
- Pruebas a nivel de sistema (3):
 - Prueba de funcionalidad: Se centran en las funciones, entradas y salidas. Por lo impracticable de probar el software para todos los juegos de datos, es necesaria la elección de buenos casos de prueba, este ejecutará el número máximo de posibilidades de entradas diferentes y debe abarcar un conjunto extenso de otros posibles casos, de manera que la ausencia o presencia de defectos, no se correspondan solo al conjunto específico de entradas de prueba, sino que los resultados sean también de conjuntos similares.

- Prueba de esfuerzo: Comprueban el desempeño y fiabilidad del sistema después de integrado.
- Prueba de rendimiento: Prueba el rendimiento del software en tiempo de ejecución dentro de un sistema integrado.
- Prueba de requerimientos: Busca discrepancias entre los requerimientos y la ejecución del software.
- Pruebas de carga: Prueba el funcionamiento del software bajo condiciones extremas.
- Pruebas de recuperación: Fuerza al software al fallo de diferentes maneras verificando que la recuperación se realice apropiadamente.
- Pruebas de seguridad: Comprueba la respuesta de los mecanismos de protección incorporados al sistema.
- Pruebas de resistencia (stress): Está diseñada para afrontar al programa en condiciones anormales.
- Pruebas a nivel de aceptación (3):
 - Prueba alfa: Se usa el software de forma natural con el desarrollador como observador del usuario.
 - Prueba beta: Es una aplicación "en vivo", del software, en un entorno que no puede ser controlado por los desarrolladores.

Existen otros tipos de pruebas que aunque no evalúan particularidades de los niveles de pruebas existentes, cubren aspectos que apuntan también hacia la obtención de un producto con un adecuado estado de calidad, estas son (3):

- Pruebas de solidez: Comprueban la correspondencia entre el tema tratado, el texto, y el resto de los medios que aparecen en la pantalla y el guión de contenido o medidas. Comprueban además la ortografía, calidad de los medios que se muestran, ya sea sonido o imágenes.
- Pruebas de documentación: Comprueban la integridad de la documentación. Se basa en la correspondencia entre el manual de usuario y lo pautado en la documentación. Se puede enfocar en dos fases. Como primera, la revisión e inspección a la documentación, para comprobar la claridad editorial. Como segundo, la prueba en vivo, que utiliza la documentación junto al uso del programa real.

- Pruebas a la interfaz gráfica de usuario: Están enfocadas a la evaluación de los atributos de la interfaz que hacen que sea más atractiva al usuario y fácil de usar.

Todas las pruebas en su conjunto permiten verificar y validar un software independientemente de las características y el entorno de desarrollo, de los recursos necesarios y disponibles para la evaluación y de factores vinculados al proceso de desarrollo del producto. Estrechamente ligado a estas aparecen las herramientas documentadas que las respaldan, pues son el medio que permite al probador conocer los detalles específicos del producto a probar. Por lo que la calidad de la documentación es un eslabón clave en la calidad del producto final.

1.3.1 Herramientas documentadas

La documentación que respalda el proceso de desarrollo de software es un fenómeno que surgió a partir del auge en la producción de software y de la necesidad de explicar lo que realiza un programador en aras de que otra persona pueda continuar lo que quedó inconcluso, o simplemente de interpretar funcionalidades del producto que no son entendible para todas las personas que interactúan con el mismo, ya sean los probadores que generalmente son personas ajenas al grupo de desarrollo del producto en cuestión o el cliente.

La documentación es una ciencia y así ha estado considerada hasta prácticamente el siglo XX. Pero también es una técnica, una herramienta para la recuperación de información. La documentación no es más que un documento, o conjunto de documentos, preferentemente de carácter oficial, que sirven para la identificación personal o para documentar o acreditar algo. (7)

Por su parte, un documento es el testimonio de la actividad del hombre fijado en un soporte perdurable que contiene información. Es un objeto que conserva la huella de la actividad humana; que sirve para dar noticia de un hecho, quedando esta noticia fijada en el objeto. Se presenta en un soporte material (piedra, pergamino, papel, cinta, disco) en el que un medio (escritura, pintura) fija el contenido, la noticia (información) (7).

En general se habla mucho de la documentación asociada a las pruebas, pero no se hace, no se le asigna presupuesto, no se mantiene y casi nunca está al día en los proyectos de desarrollo de software. Lo importante es la disponibilidad de la documentación que se necesita en el momento que se necesita. Muchas veces se hace porque hay que hacerla y se escribe con pocas ganas, textos en extremo largos, además de que el documentador está convencido de estar haciendo un trabajo inútil. A veces se peca por exceso y otras por defecto. En ocasiones se olvida que el mantenimiento también debe llegar a la documentación.

La mayor parte de los errores descubiertos durante las pruebas, son errores de documentación, ya sea de diseño o de requerimientos, son defectos que fueron introducidos antes de la primera línea de código. Las pruebas a la documentación comprueban la integridad de la misma. Se basan en la correspondencia de entre el manual de usuarios y lo pautado en la documentación. Se puede enfocar en dos fases. La primera como la revisión e inspección de la documentación, para comprobar la claridad editorial. Como segundo momento, la prueba en vivo, que utiliza la documentación junto al programa real. Esta última se puede realizar usando técnicas análogas a métodos de caja negra (3).

Diversos estudios han demostrado que la documentación de baja calidad, obsoleta o inexistente es una de las mayores causas de defectos en el desarrollo y mantención de software. Se puede decir entonces que la documentación es un componente clave en la calidad de software y el mejoramiento del proceso de software, teniendo gran impacto en la calidad del producto final.

En la actualidad el uso de la evaluación de la documentación en los procesos de desarrollo de software se está poniendo en práctica con éxito en el amplio mercado del software, pues las empresas productoras están reconociendo la necesidad de desarrollar software con calidad. Dentro de las herramientas documentadas existentes, se hará hincapié en los casos de prueba, las historias de usuario, los manuales de usuario y las listas de chequeo puesto que son las entradas al proceso de pruebas del GCF.

Casos de Prueba

En la ingeniería de software, los casos de prueba (CP) son un conjunto de condiciones o variables bajo las cuales se determinará si los requisitos de una aplicación son parciales o completamente satisfactorios (3).

La IEEE define los CP como al conjunto de entradas, condiciones de ejecución y resultados esperados, desarrollados para un objetivo particular como, por ejemplo, ejercitar un flujo de un programa o verificar el cumplimiento de un determinado requisito (4).

Cada CP está asociado a un escenario de un caso de uso en particular y extiende o amplía su descripción, permitiendo validar si el software desarrollado cumple con las especificaciones descritas en los casos de uso. Para su elaboración se necesita de los documentos de requisitos, arquitectura o de cualquier otro que muestre lo desarrollado y lo susceptible a tener errores.

Los CP de tres bloques fundamentales:

- Propósitos de ejecución de la prueba.
- Pasos de ejecución de las prueba.
- Resultados que se espera.

Abarcar de forma explícita todos estos bloques puede parecer un trabajo laborioso y tedioso, lo cual torna aburrida la fase de pruebas. A pesar de esto, es importante no incurrir en este error pues se caería en una falta mayor, subestimar una tarea imprescindible dentro del desarrollo del software: las pruebas (8). Los CP incluyen una descripción de la funcionalidad que se va a probar, la cual puede ser tomada de los casos de uso o de los requisitos, y la preparación necesaria para asegurarse de que la prueba pueda ser realizada.

Resulta vital el desarrollo de CP completos, y no solamente de lo que parezca correcto, pues esto devalúa la calidad final de las pruebas, y por consiguiente la del producto en general. Es necesario que los diseñadores estén dotados de un alto sentido de creatividad, de manera que puedan abarcar la máxima probabilidad de detección de errores con la menor cantidad de esfuerzo, recursos y tiempo disponible (3).

Existen múltiples técnicas para medir o asegurar la calidad de un conjunto de CP, como las mediciones de cobertura que alcanzan las pruebas, la cantidad de fallos encontrados por la pruebas, entre otras, sin embargo, todas técnicas no contribuyen con sus mediciones a la evaluación de la calidad del conjunto de CP sino que simplemente son intentos de mejorar la calidad de forma práctica (9).

Listas de Chequeo

Las listas de chequeo (LC) constituyen una herramienta para el control de riesgos cuya función básica se centra en la detección de condiciones en las que pueda colapsar un software. Se aplican para obtener un resultado cuantitativo de la calidad con que cuenta el producto a partir de los aspectos que se deseen medir en cada caso, el contenido de las mismas puede variar según las características del producto que se va a probar (10).

Una LC es un formulario de preguntas, las cuales dependen del objetivo para el que son usadas. Las preguntas que conforman la LC no deben ser más de doce, si se hace muy extensa, el probador puede perder el interés en el trabajo que realiza según avanza, y el resultado que se obtendría al final estaría plagado de errores. Además, se debe describir cada aspecto a revisar de forma detallada, a fin de facilitar su comprensión por parte de la persona que la va a utilizar, además que la evaluación del mismo no debe implicar cálculos ni operaciones complejas, sino que debe ser fácil de aplicar.

Las respuestas a las preguntas de la LC se pueden dar de forma directa o mediante la realización de CP; en este caso, las respuestas dependerán de los resultados obtenidos a través de los mismos. Es importante aclarar que una pregunta puede ser respondida por más de un CP.

La LC debe cubrir las áreas de problemas más comunes. Su objetivo es examinar cuidadosamente las áreas importantes y permite considerar posteriormente las mejoras que pueden ser planificadas luego de su aplicación. Las LC no resuelven los problemas, pero son un paso significativo hacia la identificación y solución de los mismos.

Historias de usuario

Son la técnica utilizada para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario (HU) es muy dinámico y flexible. Cada HU es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas y los probadores emplearlas para revisar el producto final (11).

Manual de Usuario

Un manual de usuario (MU) es un documento de comunicación técnica que busca brindar asistencia a los sujetos que usan un sistema. Más allá de su especificidad los autores deben apelar a un lenguaje ameno y simple para llegar a la mayor cantidad posible de receptores (12).

Este tipo de publicaciones brinda la información necesaria para que un usuario pueda utilizar un determinado producto o servicio. Generalmente este documento está escrito por los escritores técnicos. Por ejemplo, si el manual está referido a un producto informático, incluirá explicaciones sobre su funcionamiento, las funciones de las teclas, las opciones disponibles, etc. (12).

Los MU suelen estar escritos en diferentes idiomas y contar tanto con texto como imágenes para facilitar la comprensión de los conceptos. Los esquemas y gráficos también son herramientas muy utilizadas en este tipo de documentos (12).

Un buen punto de partida para un MU es pensar que la persona que lo va a utilizar no tiene el más mínimo conocimiento sobre el producto al que se le realiza. Por tanto a la hora de confeccionarlo es importante tener en cuenta diversos pasos a seguir para garantizar la calidad del mismo. Por ejemplo (12):

- ¿De qué trata el documento y quien lo elaboró?
- ¿Para qué sirve? ¿De qué habla?
- ¿Qué se necesita para poder instalar o usar el producto en cuestión?
- Una explicación detallada del funcionamiento del producto.

Es importante tener en cuenta que es necesario escribirlo de tal manera que cualquier persona sea capaz de entenderlo con la menor dificultad posible. Además, se deben especificar de forma detallada todos los pasos a realizar para la utilización del producto, así como el alcance y las limitaciones del mismo (12).

1.3.2 Pruebas Exploratorias

Dentro de las técnicas de pruebas, esta investigación tiene un marcado interés en las pruebas exploratorias. Estas no son más que una búsqueda táctica de las averías y defectos del software, se

refieren a ejecutar la prueba a medida que se piensa en ellas sin gastar demasiado tiempo en preparar o explicar las pruebas, confiando en los instintos (13).

Las pruebas exploratorias se definen como el aprendizaje, el diseño de casos de prueba y la aplicación de las pruebas a la vez. En otras palabras, es una técnica de pruebas en la cual quien prueba, controla activamente el diseño de las mismas mientras son realizadas. El objetivo principal de las pruebas exploratorias es descubrir cómo el software trabaja realmente y cómo manejará situaciones fáciles y difíciles por lo que siempre se debe llevar un control de los resultados obtenidos. Los resultados de las pruebas exploratorias no son necesariamente distintos de lo que se obtienen luego de la aplicación de una prueba con diseño previo (13).

Las pruebas exploratorias se pueden aplicar en cualquier situación donde no sea obvio cuál es la próxima prueba que se debe realizar. También son adecuadas cuando se quiere obtener retroalimentación rápida de cierto producto o funcionalidad, se necesita aprender el producto rápidamente, se quiere investigar y aislar un defecto en particular, conocer el estado de un riesgo en particular o se quiere evaluar la necesidad de diseñar pruebas para esa área (13).

El éxito de estas pruebas depende de la habilidad del probador para inventar situaciones que exploten al máximo las funcionalidades del software así como para encontrar defectos. Mientras más conocimientos posean los probadores del funcionamiento del software, mejor se realizarán estas pruebas. Al realizar la prueba exploratoria, no hay resultados previstos exactos; es el probador quien decide la calidad de la misma, investigando críticamente los resultados obtenidos (13).

Una de las ventajas de las pruebas exploratorias es que el probador puede emplear razonamiento deductivo de acuerdo con los resultados de las pruebas anteriores para de esta forma, encaminar el proceso de pruebas a seguir a partir de ese momento (13).

1.4 Evaluación de la calidad

Las mediciones que se realizan sobre la calidad pueden tener distintos objetivos, dependiendo de la situación en que se encuentre el proyecto y cuán aplicable sea la técnica de evaluación que se emplea.

Los modelos de análisis y diseño no se pueden probar en el sentido convencional, ya que no pueden ejecutarse. Sin embargo se pueden utilizar revisiones técnicas formales y otras técnicas para examinarlos. Para la realización de estas pruebas, aunque no se realizan de forma simultánea, se aplican las mismas técnicas que permiten descubrir errores dentro del contexto de la sintaxis semántica y pragmática de los modelos generados en estas etapas.

De acuerdo con Boris Beizer (6), Norman Fenton, Shari Pfleeger (14), Gordon Schulmeyer y James McManus (15), las técnicas para la estimación de la calidad se pueden categorizar de diversas maneras:

- Técnicas Estáticas: Se aplican a la documentación del proyecto, sin ejecución del producto. Incluyen técnicas analíticas e intensivas en personal.
- Técnicas analíticas: Se aplican a diferentes partes del producto de manera manual o automatizada. Pueden identificar defectos directamente, pero normalmente son parte de otras técnicas. Esta categoría también incluye los métodos formales.
- Técnicas intensivas en personal: Incluyen revisiones y auditorías, utilizan técnicas analíticas y pruebas.
- Técnicas dinámicas: Incluyen técnicas de prueba, pero también la simulación, el chequeo de modelos y la ejecución simbólica pueden ser consideradas dinámicas.
- Técnicas de prueba: Incluye las pruebas de aceptación y la evaluación de los artefactos de pruebas.

Particularmente, las técnicas de evaluación de la arquitectura de software permiten al arquitecto realizar mediciones sobre atributos de calidad. Clements (13) clasifica las técnicas de evaluación de la arquitectura en:

- Inquisitivas: Se destaca el uso de cuestionarios, listas de chequeo y los escenarios de prueba.
- De medición: Son utilizadas para responder interrogantes específicas sobre atributos de calidad determinados. Utilizan instrumentos como los lenguajes de descripción arquitectónica y las métricas, que son interpretaciones cuantitativas sobre mediciones observables particulares realizadas sobre la arquitectura. Son utilizadas para medir la complejidad del sistema, determinar la propensión de fallas de los módulos, etc.

Esta categorización de técnicas para la evaluación de la calidad, lleva a establecer los siguientes criterios para la selección de las técnicas más apropiadas, con base en su clasificación, objetivos y recursos para aplicarlas:

- Estática, Analítica, Inquisitiva y las mediciones cualitativas, son aplicables a las etapas tempranas del proceso, aplicables a especificaciones de análisis, diseño y pruebas y medir atributos de calidad interna.
- Dinámica, Inquisitiva y las mediciones cualitativas, se pueden aplicar en etapas tempranas del proceso, aplicables al diseño y a los atributos de calidad interna.
- Dinámica, Pruebas y las mediciones cuantitativas son aplicables en etapas tardías del proceso, se pueden aplicar al código y la integración y a los atributos de calidad externa.
- Dinámica, Inquisitiva y mediciones cuantitativas, son aplicables en etapas tardías del proceso, al código y la integración, así como a la medición de atributos de calidad externa.

En el desarrollo de la propuesta de solución, se utilizarán las técnicas estáticas para la estimación de la calidad puesto que se aplican a la documentación sin necesidad de ejecutar el producto mediante las listas de chequeo y la utilización de las mismas propuesta por las técnicas inquisitivas para la evaluación de la arquitectura.

1.4.1 Métricas de Software

Las medidas y las métricas son herramientas muy útiles para el control de los procesos. Un proceso de medidas que acompañe al proceso de software y se integre con él, ayuda en la interpretación, control y mejora de cada una de las actividades que se llevan a cabo dentro del mismo. Las medidas y las métricas han de ser diseñadas y dirigidas para soportar los objetivos de la organización para las que son desarrolladas.

La IEEE define las métricas como una medida cuantitativa del grado en que sistema, componente o proceso posee un atributo dado (4). Se dice que la medición es esencial si se desea conseguir realmente la calidad en software, es por eso que existen diferentes tipos de métricas, para poder evaluar, mejorar y clasificar al software final, en donde serán manejadas dependiendo del entorno de desarrollo del software al cual pretendan enfocarse.

Cuando se planifica un proyecto se tiene que obtener una estimación del costo y esfuerzo humano requerido, esto se obtiene por medio de las mediciones de software que se utilizan para recolectar los datos cualitativos a cerca del software y sus procesos para aumentar la calidad.

Las métricas ayudan a entender tanto el proceso técnico que se utiliza para desarrollar el producto como el producto en sí. Son un buen método para entender, monitorizar y controlar, predecir y probar el desarrollo del software y los proyectos de mantenimiento. Las métricas son la maduración de una disciplina, según Pressman permiten (3) :

- Evaluación de los modelos de análisis y diseño.
- Proporcionan una indicación de la complejidad de diseños procedimentales y de código fuente.
- Ayudan al diseño de pruebas más efectivas.

Por estas razones, propone un proceso de medición que se puede caracterizar por cinco actividades (3):

1. Formulación: Obtención de medidas y métricas del software apropiadas para la representación del software en cuestión.
2. Colección: Mecanismo empleado para acumular datos necesarios para obtener las métricas formuladas.
3. Análisis: Cálculo de las métricas y aplicación de las mismas.
4. Interpretación: Evaluación de los resultados de las métricas en un esfuerzo por conseguir una visión interna de la calidad de la representación.
5. Realimentación: Recomendaciones obtenidas de la interpretación de métricas técnicas transmitidas al equipo de software.

La medición es una disciplina relativamente joven, y no existe un consenso general sobre la definición exacta de los conceptos y la terminología que maneja. Inicialmente puede parecer que la medición del software es algo evidente. Después de todo es lo que permite cuantificar y por consiguiente gestionar de forma más efectiva. Pero la realidad es muy diferente, frecuentemente la medición lleva a gran controversia y discusión.

- ¿Cuáles son las métricas apropiadas al proceso y al producto?

- ¿Cómo se deben usar los datos que se recopilan?
- ¿Es bueno usar medidas para comparar gente, procesos o productos?

Estas preguntas y muchas más siempre surgen cuando se intenta medir algo que no ha sido medido en el pasado. (16)

En general la medición persigue tres objetivos fundamentales:

- Ayudar a entender qué ocurre durante el desarrollo y mantenimiento.
- Permite controlar que es lo que ocurre en los proyectos.
- Se puede mejorar los procesos y los productos. (17)

Las métricas de software es un término que se asigna a un amplio rango de actividades diversas, por ejemplo:

- Medidas y modelos de estimación de coste y esfuerzo.
- Modelos y medidas de productividad.
- Aseguramiento y control de calidad.
- Recogida de datos.
- Medidas y modelos de calidad.
- Modelos de fiabilidad.
- Modelos y evaluación de ejecución.
- Complejidad computacional o algorítmica. (17)

Hay varias razones para medir un producto:

- Para indicar la calidad del producto.
- Para evaluar la productividad de la gente que desarrolla el producto.
- Para evaluar los beneficios en términos de productividad y de calidad, derivados del uso de nuevos métodos y herramientas de la ingeniería de software.
- Para establecer una línea de base para la estimación.
- Para ayudar a justificar el uso de nuevas herramientas o de formación adicional. (16)

Las mediciones pueden englobarse en dos categorías: medidas directas y medidas indirectas.

1. Medidas directas: En el proceso de ingeniería se encuentran el costo y el esfuerzo aplicado, las líneas de código producidas, velocidad de ejecución, el tamaño de memoria y los defectos observados en un determinado período de tiempo.
2. Medidas indirectas: Se encuentra la funcionalidad, calidad, complejidad, eficiencia, fiabilidad, facilidad de mantenimiento, etc.

Como parte del software hay tres clases de entidades cuyos atributos pueden ser medidos:

- Procesos: Son actividades de software que normalmente conllevan el factor tiempo. Atributos internos interesantes: el tiempo (duración de esfuerzo), el esfuerzo (asociado al proceso), y el número de incidentes de un tipo específico que se dan durante el proceso (por ejemplo, el número de errores de requisitos encontrados durante la construcción de la especificación).
- Productos: Son entregables, artefactos o documentos generados en el ciclo de vida del software. Ejemplos de atributos externos: la fiabilidad del código, cuán entendible es un documento de especificación, qué tanto se puede mantener el código fuente, etc. Ejemplos de atributos internos: la longitud, la funcionalidad, la modularidad o corrección sintáctica de los documentos de especificación.
- Recursos: Son todos aquellos elementos que hacen de entrada a la producción de software. Por ejemplo: el personal, los materiales, las herramientas y los métodos. Un atributo interesante es el coste. En el caso del personal, además del coste se suele medir la productividad. (17)

Últimamente ha aparecido un gran número de métricas para capturar atributos del software de forma cuantitativa. Sin embargo muy pocas métricas han sobrevivido a la fase de definición y se usan en la industria, esto se debe a múltiples problemas, entre ellos:

- Las métricas no se definen siempre en el contexto del objetivo de interés industrial que se pretende alcanzar.
- Las definiciones de métricas no siempre tienen en cuenta el entorno o contexto en que serán aplicadas.

- No siempre es posible realizar una validación teórica adecuada de la métrica porque el atributo que se quiere medir no siempre está bien definido.
- Un gran número de métricas nunca se ha validado empíricamente.

Esta situación ha conducido a cierto grado de ambigüedad en las definiciones, propiedades y asunciones de las métricas, haciendo que el uso de las mismas sea difícil, la interpretación peligrosa y los resultados contradictorios. Para evitarlo es necesario contar con un método de definición de métricas y con una base para su formalización. (17)

Métricas para la medición de Software

Las métricas del software son las que están relacionadas con el desarrollo del mismo, ya sea funcionalidad, complejidad, eficiencia, etc. (17)

- **Métricas Técnicas:** Se centran en las características de software por ejemplo: la complejidad lógica, el grado de modularidad. Mide la estructura del sistema, el cómo está hecho.
- **Métricas de Calidad:** Proporcionan una indicación de cómo se ajusta el software a los requisitos implícitos y explícitos del cliente. Es decir cómo medir para que el sistema se adapte a los requisitos que pide el cliente.
- **Métricas de Productividad:** Se centran en el rendimiento del proceso de la ingeniería del software. Es decir que tan productivo va a ser el software que se va a diseñar.
- **Métricas orientadas a la persona:** Proporcionan medidas e información sobre la forma que la gente desarrolla el software de computadoras y sobre todo el punto de vista humano de la efectividad de las herramientas y métodos. Son las medidas que se hacen sobre el personal que hará el sistema.
- **Métricas orientadas al tamaño:** Es para saber en qué tiempo se va a terminar el software y cuantas personas se van a necesitar. Son medidas directas al software y el proceso por el cual se desarrolla, si una organización de software mantiene registros sencillos, se puede crear una tabla de datos orientados al tamaño.
- **Métricas orientadas a la función:** Son medidas indirectas del software y del proceso por el cual se desarrolla. Las métricas orientadas a la función se centran en la funcionalidad o utilidad del programa.

Métricas para la medición de la Documentación

Como ocurre con la cuantificación de la calidad del software, la de un documento también podría realizarse si se consigue identificar un conjunto suficientemente representativo de variables de medida para ello. En este sentido hay que diferenciar, por una parte, la medida de la calidad del texto incluido en un documento, la calidad de la interfaz del documento con el usuario, la calidad de la estructura del documento, sobre todo en documentos hipertexto o hipermedia, y, por otro lado, la calidad de los diagramas o modelos que aparecen en ellos como consecuencia de haber aplicado alguna metodología de desarrollo de software.

- **Métodos basados en las pruebas de CLOZE:** Su nombre proviene de closure construct y hace referencia a la característica humana que permite completar caracteres incompletos según el contexto. El principal método de este tipo es el "procedimiento de Taylor", utilizado inicialmente en el ámbito periodístico (años cincuenta). Después se aplicó en el ámbito de la literatura y traducción para comprobar la comprensibilidad de los textos. También ha sido utilizado en la documentación de proyectos de software (18). Este método consiste en eliminar palabras del texto original a intervalos regulares, ofrecer este texto incompleto a un cierto número de personas relacionadas con el tema, recoger las opiniones de estas personas respecto a cuáles podrían ser las palabras que faltan, contabilizar el número de palabras acertadas y calcular el coeficiente de comprensibilidad del texto (entre 0 y 1) como el cociente del número de aciertos y el total de palabras eliminadas.
- **Métodos de formalización de la comprensibilidad:** Se caracterizan por determinar la comprensibilidad y legibilidad de los textos cuantificando formalmente atributos léxicos y sintácticos. Son los métodos más científicos, que obtienen índices de comprensibilidad a partir de diferentes fórmulas matemáticas. (19).
- **Métodos de ordenación/estructuración de la información:** Con estos métodos se trata de "medir" la adecuada disposición de los contenidos según un orden secuencial, desde aspectos generales hasta aspectos específicos. Este tema ha estado bastante olvidado hasta la aparición de los documentos hipertexto e hipermedia, cuyas enormes posibilidades de navegación por su contenido ha planteado la necesidad de establecer métricas para evaluar la calidad de los enlaces hipermediales (20). Diversos autores han propuesto un método formal de cuantificación de la calidad basándose en el análisis de la estructura jerárquica de los hipertextos, midiendo, entre

otros, la compactación (lo intrincado de las conexiones) y la disposición (el grado de organización, que se refiere al número de nodos que deben ser leídos antes que otros) del documento.

- **Métodos basados en el medio de presentación:** Se basan en cuantificar la calidad de la forma de presentar la información al lector. Esto tiene una gran importancia en el caso de la documentación hipertextual, cuya lectura se realiza a través de una interface de usuario cuya calidad debería poder valorarse, ya que influirá en la atención que se prestará en la lectura y, por ende, en la comprensión del contenido. En este sentido existen autores que han analizado los parámetros que influyen en la comprensión de un hipertexto (20) teniendo en cuenta el proceso cognitivo que se produce en el lector cuando lo utiliza, estos parámetros son el control, la participación y la síntesis (21) (22).
- **Métodos intuitivos basados en la experiencia:** Son métodos sin justificación científica, pero que han adquirido un cierto nivel de aceptación, algunos promedian la opinión de un cierto número de lectores respecto a la comprensión individual de palabras (o frases), verbos activos, nombres personales y sustantivos abstractos.
- **Métodos basados en impresiones subjetivas:** Consiste en cuantificar las opiniones subjetivas de expertos en forma de grado de satisfacción con respecto a atributos como simplicidad, organización, brevedad, estimulación, etc.
- **Métodos basados en la relación texto-lector:** Consideran la comprensión como una interrelación entre el lector y el texto. El principal método de este tipo es el propuesto por Groeben (19), que construye una teoría del conocimiento en la que la comprensión de un texto es vista como la "construcción cognitiva de una estructura semántica". Groeben diferencia, por primera vez, entre la "comprensión individual de un texto", en la que se considera la influencia de los atributos del lector en el proceso de comprensión, y la "comprensibilidad intercultural de un texto", que describe la influencia de los atributos del texto en tal proceso. Esta comprensibilidad se evalúa a través de cuatro dimensiones: la estructura cognitiva del texto, la simplicidad lingüística y la brevedad semántica, la redundancia, y la conflictividad en la estimulación cognitiva (incongruencias, incoherencias, complejidad).

Ninguno de los métodos para la medición de la documentación se ajusta a la propuesta de solución que se desarrollará en capítulos posteriores de la presente investigación, puesto que se enfocan en la

comprensión de la información o en la presentación de la misma, nunca se acercan a la comprobación de la completitud, actualización o correctitud de los documentos, siendo estos los aspectos más importantes para elevar la calidad de las herramientas documentadas que se emplean como entrada al proceso de pruebas del GCF.

1.5 Procesos de prueba de software

Proceso

Una definición sencilla de proceso es: Serie de acciones que conducen a un final. Esta definición coincide aparentemente con la idea que las personas tienen sobre un proceso, pero deja muchas preguntas abiertas. ¿El proceso es la forma en que la organización opera, o es la forma en que un diseñador diseña, produce código o prueba el código? ¿El proceso se refiere a administración, ingeniería o ambas? ¿El proceso implica demasiada documentación y abstiene de desarrollar el producto objetivo? (23)

La respuesta a estas preguntas puede variar dependiendo de la perspectiva. Sin embargo, siempre que para alcanzar algún fin se necesite ejecutar una serie de acciones y estas tengan cierto orden, dependencias, roles responsables, resultados, tiempos de ejecución y herramientas de apoyo, se estará hablando de procesos que pueden ser predefinidos y personalizados (23).

Es importante tener en cuenta que un proceso no es lo mismo que un procedimiento, este es el conjunto de reglas e instrucciones que determinan la manera de proceder o de obrar para conseguir un resultado. Un proceso define qué es lo que se hace y un procedimiento cómo hacerlo (23).

No todas las actividades que se realizan son procesos. Para determinar si una actividad realizada por una organización es un proceso o subproceso, debe cumplir los siguientes criterios (23):

- La actividad tiene una misión o propósito claro.
- La actividad contiene entradas y salidas, se pueden identificar los clientes, proveedores y producto final.
- La actividad debe ser susceptible de descomponerse en operaciones o tareas.
- La actividad puede ser estabilizada mediante la aplicación de la metodología de gestión por procesos (tiempo, recursos, coste).

- Se puede asignar la responsabilidad del proceso a una persona.

Procesos de Pruebas de Software

Según aumenta la complejidad de los sistemas de software y la demanda de la calidad se hacen necesarios procesos y métodos que permitan obtener buenos conjuntos de pruebas de sistema. Un proceso de pruebas puede definirse de manera general como un conjunto de actividades y entregables que se siguen para realizar pruebas a un software, garantizando un producto confiable, sin errores y con la mayor calidad posible. Se recomienda su integración dentro del propio desarrollo del producto y no de manera aislada (3).

Consta generalmente de cuatro fases: diseño de pruebas, codificación, ejecución y análisis de resultados. Los procesos toman como punto de partida los documentos relacionados con las pruebas, a partir de ellos genera los resultados y construye las pruebas. Un proceso de pruebas es enfocado sobre la lógica interna del software y las funciones externas (3).

La gran mayoría de las metodologías de pruebas distinguen 5 actividades principales en el proceso de pruebas, aunque con nombres diferentes. Estas son:

- Planificación o Estrategia: qué se va a hacer, cuándo, cómo y quién lo hará, que es necesario satisfacer para llevarlo a cabo.
- Diseño de pruebas: qué pruebas se van a hacer y en qué consiste cada una (condiciones, pasos, datos, resultados esperados)
- Construcción de pruebas: especialmente en el caso de pruebas automáticas, es necesario materializar el diseño en determinados artefactos (programas, scripts de herramientas de ejecución automática).
- Ejecución: realización de las pruebas propiamente dichas. Todo lo anterior es la preparación para esta fase.
- Evaluación: análisis de los resultados obtenidos de cara a proporcionar un diagnóstico del sistema bajo prueba y del propio proceso de prueba.

Adicionalmente, hay otros procesos necesarios, que las metodologías suelen desarrollar:

- Seguimiento: para asegurar que las actividades se realizan de acuerdo a lo previsto o, en caso contrario, se toman las medidas adecuadas.
- Gestión de la configuración: para gestionar y controlar productos de entrada y salida de las distintas actividades, y sus relaciones entre ellos y con el sistema bajo prueba.

Proceso de Pruebas de Software CALISOFT

El proceso de pruebas de CALISOFT está enfocado a la liberación de productos, el mismo se inicia cuando se realiza una solicitud de pruebas, la cual se revisa y se le informa al proyecto que la realiza si la misma se acepta o no. En caso de ser aceptada se procede a realizar la reunión de inicio, donde se elabora el plan de pruebas (24).

Posteriormente se procede a montar el entorno de pruebas y diseñar y/o revisar las pruebas, luego se realizan pruebas exploratorias iniciales, donde se determina si se cumplen los criterios de criticidad o no, en caso de que se cumplan las pruebas quedan suspendidas, de lo contrario se comienzan a desarrollar las iteraciones de pruebas. Luego de cada iteración se evalúan nuevamente los criterios de criticidad y en caso de no cumplirse se envía la notificación de liberación al proyecto y se sube la versión liberada al repositorio central. Las actividades anteriormente descritas se evidencian de forma más clara a través del siguiente diagrama (24):

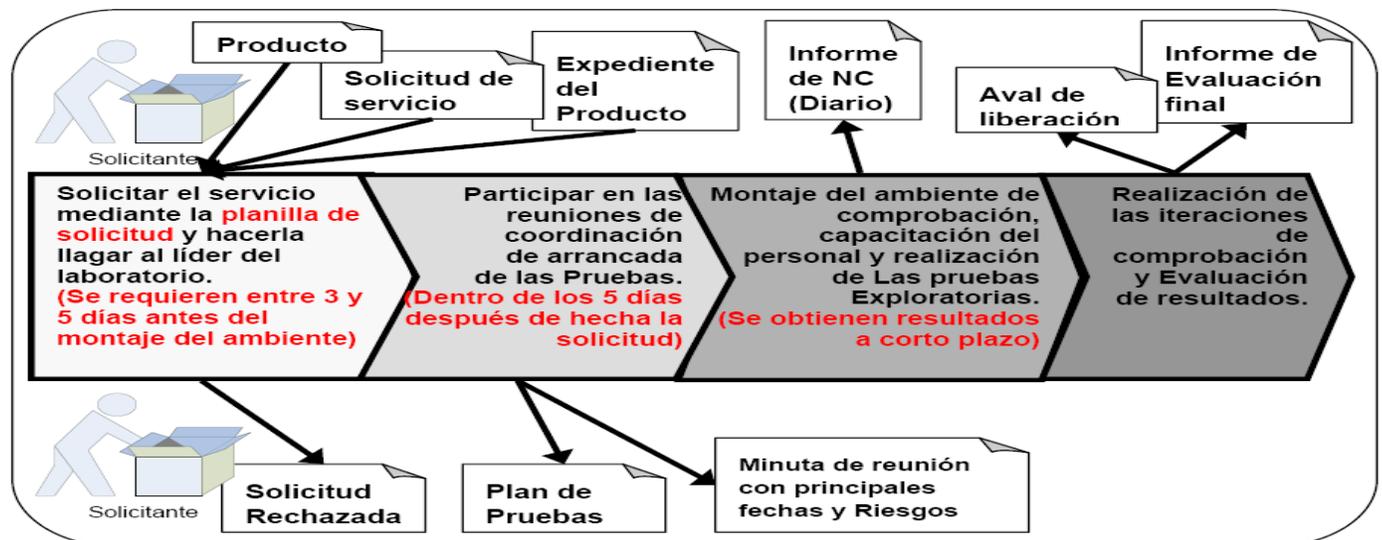


Imagen 1. Diagrama del servicio de liberación de software de CALISOFT

Proceso de Pruebas de Software GCF

El proceso de pruebas de software del GCF está enfocado en las pruebas funcionales de sistema, puesto que este es el servicio prestado por dicho grupo. Este se desglosa en varios subprocesos para garantizar una mayor calidad en las actividades que engloba como se evidencia en la imagen que se muestra a continuación, posteriormente se realizará una breve descripción de los mismos para conocer en qué consisten y cuál es su objetivo dentro del proceso de pruebas de software del GCF (1):



Imagen 2. Mapa de procesos del Servicio de pruebas del GCF.

- Procesos Estratégicos: Aquellos que proporcionan directrices a todos los demás procesos y son realizados por la dirección o por otras entidades. Se suelen referir a las leyes, normativas etc. aplicables al servicio y que no son controladas por el mismo.
 - Evaluación preventiva de la documentación.
 - Auditorías y Revisiones periódicas
- Procesos Fundamentales: Atañen a diferentes áreas del servicio y tienen impacto en el cliente creando valor para éste. Son las actividades esenciales del servicio, su razón de ser.
 - Pruebas de Software a Nivel de Sistema.

- Procesos de Soporte: Procesos que apoyan la realización de los Procesos Fundamentales.
 - Asesoramiento sobre temas relacionados con el proceso de pruebas:
 - Elementos organizativos para el proceso de prueba.
 - Capacitación para la elaboración de documentos como:
 - ✓ Plan de Pruebas.
 - ✓ CP.
 - ✓ LC.
 - Asesoramiento para la utilización de la herramienta de automatización y gestión de pruebas (TRAC).

Propone además seis tareas fundamentales, de su organización, cumplimiento y control, depende en gran medida el éxito de una prueba de software (1) :

- Atención a las solicitudes de Pruebas (o Inicialización).
 - Revisión de las solicitudes.
 - Comunicación de aceptación o rechazo de las solicitudes.
 - Asignación de prioridades.
 - Definición de fechas tentativas para realización de las Reuniones de Inicio.
- Planificación de Pruebas.
 - Realización de la Reunión de Inicio de las pruebas.
 - Gestión de acuerdos tomados en la Reunión de Inicio.
 - Identificación de los posibles riesgos a tener en cuenta durante el período de pruebas.
 - Aseguramiento de los recursos humanos que ofrecerán el servicio.
 - Creación de las condiciones técnicas necesarias para la realización del servicio.
- Capacitación para la ejecución del servicio.
 - Estudio de la documentación para la realización del servicio.
 - Familiarización con el tipo de pruebas a realizar mediante el intercambio directo con miembros del Equipo de Desarrollo.
 - Elaboración de LC, Diseño y/o Automatización de Pruebas.
 - Revisiones de LC, Diseño y/o Automatización de Pruebas.
 - Automatización de procedimientos de pruebas.

- Ejecución de las Pruebas.
 - Realización de las pruebas.
 - Informe el comportamiento de las no conformidades durante cada iteración.
 - Realización de las Reuniones de Conciliación luego de cada iteración.
 - Seguimiento y gestión de los acuerdos tomados en las Reuniones de Conciliación.
 - Cierre de Iteraciones.
- Evaluación de las Pruebas.
 - Análisis del comportamiento de las no conformidades por cada iteración.
 - Análisis estadístico de las no conformidades por cada evaluación.
 - Evaluación de la calidad del trabajo de los miembros del GCF.
- Cierre de las Pruebas.
 - Realización del resumen del servicio ofrecido.
 - Reunión de cierre de pruebas.
 - Comunicación de la evaluación del trabajo realizado a los miembros del GCF.

Las tareas anteriormente descritas se evidencian a través de la siguiente imagen:



Imagen 3. Tareas del proceso para la realización de Pruebas de Software a nivel de sistema del GCF.

La presente investigación forma parte del subproceso estratégico Evaluación preventiva de la documentación. El mismo persigue el objetivo de garantizar que los documentos de entrada al proceso

fundamental que se desarrolla (Pruebas de software a nivel de sistema), estén actualizados, es decir que la aplicación se corresponda con la herramienta documentada a utilizar, que estén completos, o sea, que todos los datos necesarios para la comprensión y utilización de la información en el proceso de pruebas aparezca recogida dentro del documento, y por último, que estén correctos, esto significa que el contenido de la documentación en cuestión, debe corresponderse con lo especificado en el documento de requisitos, o en los casos de uso.

1.5.1 Metodologías para el diseño y mejora de procesos

Si se incluye dentro de la cultura organizacional del GCF la coexistencia controlada de la mejora continua y de la innovación, se puede garantizar el camino a la competitividad y por supuesto, al logro de los objetivos.

La mejora continua, implica entender y trabajar en la cadena de valor Proveedores-Organización-Cliente, y directamente en los procesos que configuran esta cadena, sumando las diferentes iniciativas de mejora. El trabajo que se desarrolla, debe ser entendido como una serie de procesos que deben ser mejorados constantemente sobre la base de (25):

- Comportamiento de equipo.
- Compromiso de mejora constante.
- Establecimiento de objetivos locales.
- Establecimiento de mecanismos de medición.
- Verificación de resultados.
- Aplicación de medidas correctivas o preventivas, de acuerdo a los resultados obtenidos, etc.

La mejora de los procesos, significa optimizar la efectividad y la eficiencia, mejorando también los controles, reforzando los mecanismos internos para responder a las contingencias y las demandas de nuevos y futuros clientes. Para mejorar los procesos, se debe considerar (25):

- Análisis de los flujos de trabajo.
- Fijar objetivos de satisfacción del cliente, para conducir la ejecución de los procesos.
- Desarrollar las actividades de mejora entre los protagonistas del proceso.
- Responsabilidad e involucramiento de los actores del proceso.

Para establecer una metodología clara para la comprensión de la secuencia de actividades o pasos que se deben aplicar para la mejora continua de los procesos, primero, el responsable del área debe saber que mejorar. Esta información se basa en el cumplimiento o incumplimiento de los objetivos locales de la organización. Por lo que en pos de establecer una secuencia de pasos para la mejora, estos serían (25):

- Definir el problema o la desviación detectada sobre los indicadores y objetivos.
- Establecer los mecanismos de medición más adecuados de acuerdo a la naturaleza del problema.
- Identificar las causas que originan el problema, determinando la más relevante, estableciendo posibles soluciones y tomar la opción más adecuada, por medio del análisis de los datos obtenidos.
- Establecer los planes de acción, e implementar la mejora.
- Controlar la mejora del proceso, efectuando los ajustes necesarios, por medio de un monitoreo constante.

La mejora continua de los procesos, alineada con el resto de los principios de la gestión de calidad, debe encaminar a la organización, al logro de la excelencia, o dicho de otra forma, alcanzar la calidad total (25).

Precisamente, la calidad total, se fundamenta en cinco principios, de los cuales la mejora continua es parte fundamental, siendo los otros cuatro (2):

- El enfoque al cliente
- El involucramiento total del personal
- La Medición y el establecimiento de objetivos
- El apoyo al esfuerzo por la calidad y el liderazgo de la dirección.

La mejora continua, significa mejorar los estándares, estableciendo a su vez, estándares más altos, por lo que una vez establecido este concepto, el trabajo de mantenimiento por la administración o por el responsable del proceso, consiste en procurar que se observen los nuevos estándares (25).

La mejora continua duradera, sólo se logra cuando el personal trabaja para estándares más altos, de este modo, el mantenimiento y el mejoramiento son una mancuerna inseparable. Por tal motivo, cuando se

efectúan mejoras en los procesos, éstas a la larga, conducirán a mejorar la calidad y la productividad, evitando así, la preocupación por los resultados (25).

A continuación se mencionan algunas de las metodologías existentes para el diseño y mejora continua de procesos:

- **SCOR** (Supply Chain Operations Reference): Permite describir las actividades de negocio necesarias para satisfacer la demanda de un cliente. El Modelo está organizado alrededor de los cinco Procesos Principales de Gestión: Planificación (Plan), Aprovechamiento (Source), Manufactura (Make), Distribución (Deliver) y Devolución (Return) (26).
- **Lean Manufacturing** (Manufactura esbelta): El objetivo es encontrar herramientas que ayuden a eliminar todos los desperdicios y todas las operaciones que no le agregan valor al producto o a los procesos, aumentando el valor de cada actividad realizada y eliminando lo que no se requiere. Este proceso busca relacionar los costos con todos los valores que el cliente percibe del producto. Por otro lado, sirve para implantar una filosofía de mejora continua que le permita a las compañías reducir sus costos, mejorar los procesos y eliminar los desperdicios para aumentar la satisfacción de los clientes y mantener el margen de utilidad. El propósito de la manufactura esbelta es serle útil a la comunidad lo cual implica estar en busca de la mejora continua (27).
- **Six Sigma**: La filosofía Six Sigma busca ofrecer mejores productos o servicios, de una manera cada vez más rápida y a más bajo costo, mediante la reducción de la variación de cualquiera de los procesos presentes en la organización. El concepto de Six Sigma provee una medición común, así como objetivos comunes, a la vez que inculca una visión común y sobre todo promueve el trabajo en equipo. Adicionalmente combina objetivos agresivos con un método y un conjunto de herramientas, que se aplican a través de todo el ciclo de vida del proceso o servicio. Existe una alta correlación entre la mejora del tiempo de ciclo y la reducción de defectos y costos. Muchas empresas utilizan el concepto de Seis Sigma para establecer un parámetro de negociación durante los procesos de negociación Cliente – Proveedor Interno. Su característica fundamental es la aparición de 3,4 defectos por millones de productos revisados (28).
- **Ruta de Calidad**: La ruta de la Calidad es un procedimiento para solucionar problemas. En los términos usados en ella, un problema se define de la siguiente manera: "Un problema es el resultado no deseado de una tarea". La solución para un problema es mejorar el resultado

deficiente hasta lograr un nivel razonable. Las causas de los problemas se investigan desde el punto de vista de los hechos y se analiza con precisión la relación causa efecto. Se evitan estrictamente las decisiones sin fundamento basadas en la imaginación o en la teoría desde un escritorio, debido a que los intentos de solucionar los problemas con base en decisiones orientan en direcciones equivocadas, lo cual lleva al fracaso o a demorar la mejora. Si diseñan y se implementan medidas que contrarresten el problema para evitar que los factores causales vuelvan a presentarse (29).

- **Teoría de Restricciones:** Es un conjunto de procesos de pensamiento que utiliza la lógica de la causa y efecto para entender lo que sucede y así encontrar maneras de mejorar. Está basada en el simple hecho de que los procesos multitarea, de cualquier ámbito, solo se mueven a la velocidad del paso más lento. La manera de acelerar el proceso es utilizar un catalizador es el paso más lento y lograr que trabaje hasta el límite de su capacidad para acelerar el proceso completo. La teoría enfatiza la dilucidación, los hallazgos y apoyos del principal factor limitante (30).
- **Sistemas Suaves:** Permite ir desde una situación no estructurada (situación problema) hacia una mejora de la situación. En otras palabras la Metodología sirve para “estructurar” (entender, comprender, interpretar, y mejorar) una situación no estructurada (31).

Por las razones anteriormente expuestas, se decide utilizar la metodología Six Sigma para la mejora y diseño del proceso, ya que la misma, a diferencia del resto, evalúa los procesos basándose en su capacidad para dar respuesta efectiva a los requisitos del cliente.

A corto plazo aporta soluciones rápidas a problemas sencillos o repetitivos; a largo plazo aporta una metodología de diagnóstico, diseño robusto, establecimientos de tolerancias, al tiempo que aporta un medio sencillo de comunicación y establecimiento de metas. Aporta herramientas de mejora: diseño de experimentos, análisis de regresión, tolerancias, diseño robusto y otros métodos sistemáticos para reducir la varianza.

1.6 Conclusiones del Capítulo

La calidad de software representa el broche de oro en la cadena de la producción de software, es por esto que no se le puede restar importancia ni ser tomada a la ligera, o como un aspecto decorativo para el producto. En el transcurso de este capítulo se analizaron diversas vías en pos de lograr una garantía de la misma, partiendo de un análisis de los procesos de pruebas que se realizan en la UCI, así como de los métodos necesarios para medir la calidad de dichos procesos.

Para realizar la evaluación de la documentación se estudiaron métricas relacionadas con la medición de documentos, pero aunque las analizadas son perfectamente aplicables al caso particular de la valoración de la calidad de la documentación de los proyectos de desarrollo de software, ninguna aporta información concluyente que permita emitir una valoración profunda a cerca del estado de los documentos. Por esta razón se hizo necesario estudiar a fondo las métricas relacionadas con la medición de software, y buscar entre las mismas alguna que permitiera ser aplicada a la medición de documentos. El análisis arrojó que algunas de las métricas para medir la calidad del software pueden fusionarse, tomando aspectos específicos de cada una de ellas y crear un sistema que permita evaluar el estado final de los documentos.

A modo de resumen se puede decir que la calidad de un proceso de pruebas va a depender de la calidad de los subprocesos que sirvan de soporte a la actividad fundamental, por lo que la calidad de la documentación que sirve como entrada a los procesos de pruebas es un fuerte indicador para garantizar el éxito del mismo. Por lo que es necesario tener en cuenta los siguientes aspectos:

- Antes de comenzar a realizar las actividades de los procesos, ya sean de prueba, evaluación, o cualquier otra categoría, es necesario hacer una evaluación previa para determinar si realmente vale la pena invertir tiempo y esfuerzo en la ejecución del mismo.
- Actualmente no existen métricas que permitan medir de forma específica la calidad de los documentos de los proyectos de desarrollo de software. Por lo que para llevar a cabo un proceso de evaluación de documentación es necesario fusionar técnicas y métricas que permitan a las partes interesadas formarse una idea del estado de los documentos.

CAPÍTULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN PROCESO DE EVALUACIÓN DE LA DOCUMENTACIÓN PARA EL SERVICIO DE PRUEBAS FUNCIONALES A NIVEL DE SISTEMA QUE OFRECE EL GRUPO DE CALIDAD DE FORTES.

2.1 Introducción

Dentro del presente capítulo se plasma la estructura y principales componentes de la solución propuesta, así como las actividades a tener en cuenta para su ejecución. Además se describe cómo debe ser el equipo encargado de llevar a cabo las evaluaciones y las principales plantillas a utilizar.

El presente Modelo está dirigido al GCF, el cual ha formado parte de la historia productiva de la Facultad 8, actual Facultad 4, pero en sus períodos iniciales, producto a la inexperiencia de su dirección para encaminar las actividades relacionadas con esta área, afrontó disímiles dificultades. En el segundo semestre del curso 2009-2010, se reorganiza la producción a nivel de Universidad con la creación de los centros de desarrollo productivo, y el Centro de Tecnologías para la Formación (FORTES) hereda el Grupo de Calidad de la Facultad 8, actual: Grupo de Calidad de FORTES. (1)

Con el impulso del cambio, un análisis parcial indicó que para elevar la calidad del centro se hace necesario redefinir los procesos de calidad actualmente implementados, aportando mejoras que se ajusten a las características del área. En tal sentido, fomentar una cultura en torno a la importancia del aseguramiento del proceso de desarrollo de software en FORTES, se convierte entonces en la misión fundamental de los servicios que de manera general y necesaria ofrece el Grupo de Calidad. (1)

Desde sus inicios, la actividad fundamental del Grupo de Calidad fue la realización de pruebas a nivel de sistema, tanto a productos de la facultad previos a liberarse, como a productos de otras facultades, siempre con la supervisión y participación de especialistas de la Dirección de Calidad de la universidad, hoy Centro Nacional de Calidad de Software, CALISOFT. El Equipo de Pruebas que respalda esta actividad, se subordina directamente a la dirección del Grupo de Calidad de FORTES. Está compuesto por estudiantes de 3ro, 4to y 5to año de la carrera de Ingeniería en Ciencias Informáticas y sigue una estructura organizacional basada en el ejercicio de cada uno de sus roles. (1)

CAPÍTULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

El Equipo de Pruebas del Grupo de Calidad de FORTES tiene la misión de alcanzar de manera efectiva, eficaz y eficiente los siguientes objetivos (1):

- Ofrecer servicios de pruebas de software a todos los proyectos productivos de FORTES y a los interesados, realizando actividades de verificación en correspondencia con los roles definidos.
- Formar profesionales competentes especializados en pruebas de software desde la práctica laboral en el área productiva de FORTES y con la dirección del Grupo de Calidad.

Además se rige por los siguientes principios:

- Todos los proyectos productivos de FORTES deben solicitar los servicios de pruebas al Equipo de Pruebas antes de solicitar las de pruebas de liberación a CALISOFT.
- Las actividades del Equipo de Pruebas deben eliminar los conflictos de intereses asociados con el hecho de permitir a los desarrolladores que prueben lo que han construido.
- Las actividades del Equipo de Pruebas deben garantizar el seguimiento de las pruebas hasta lograr los objetivos del cliente.
- El Equipo de Pruebas debe brindar asesoramiento a los Proyectos Productivos.
- El Equipo de Pruebas debe planificar las pruebas mucho antes de comenzarlas. (1)

Sus principales clientes son:

- Proyectos Productivos de FORTES que soliciten cualquier servicio de pruebas de software.
- Estudiantes de 5to año de la carrera de cualquier centro de la universidad que soliciten servicios de pruebas de software.
- Entidades externas a FORTES que soliciten servicios de pruebas de software. (1)

2.2 Descripción del Proceso de Evaluación

La presente investigación propone un proceso de evaluación para la documentación de los proyectos productivos de FORTES, cuyo objetivo principal es alcanzar la máxima calidad posible de la documentación, teniendo en cuenta las variables ajenas que inciden en la misma. Este proceso está encaminado a garantizar que los documentos que se sirven de entrada al proceso de pruebas de sistema del GCF, se encuentren correctos, completos y actualizados para ser utilizados en los momentos que se hagan necesarios.

CAPÍTULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

Este proceso incluye 6 tareas o actividades fundamentales, las cuales son:

1. Ejecución de Pruebas Exploratorias
2. Planificación de las revisiones
3. Capacitación del equipo de trabajo
4. Ejecución de las revisiones
5. Evaluación de las revisiones
6. Cierre de las revisiones

Es importante aclarar que antes de aplicar por primera vez el proceso, es necesario impartir una capacitación a las personas encargadas de realizar la documentación de proyectos del centro, a fin de pactar cómo se deben hacer a partir de ese momento, porque si se realizan las pruebas exploratorias desde antes de esta actividad, la cantidad de no conformidades encontradas sería elevada, puesto que los documentos no estarán hechos de la forma requerida para garantizar el éxito del proceso de revisiones. Además, no es correcto realizar una evaluación sin antes informar los parámetros para la misma.

A grandes rasgos, la capacitación previa debe instruir a cerca de:

- Forma en que los documentos deben ser desarrollados y cómo brindarles mantenimiento.
- Cuál debe ser el contenido, estructura y apariencia de los documentos.
- Forma de identificar y almacenar los documentos.
- Estilos, letras, usos de logos y demás cuestiones relacionadas con el formato que debe tener la documentación.

De la responsabilidad, organización, compromiso y seriedad, con que los proyectos del FORTES asuman estas tareas, así como el cumplimiento y control por parte del equipo de trabajo encargado de realizarlas, dependerá en gran medida el éxito de la evaluación, así como la mejoría en la calidad de las pruebas de sistema que se realizarán a los productos posteriormente.

2.2.1 Ejecución de Pruebas Exploratorias

El objetivo principal de esta actividad es aportar los elementos necesarios para obtener una aproximación a la cantidad de no conformidades existentes y con qué aspecto se relacionan las mismas (actualización, corrección, completitud u otro que pueda resultar relevante para la realización de las pruebas de software), de esta forma se puede hacer un análisis de los resultados obtenidos y determinar si los documentos pueden ser revisados. Esta actividad es de suma importancia para el proceso de evaluación; le permite al equipo de pruebas preconcebir una idea del estado general de los documentos que se revisarán posteriormente.

Con este fin se evalúan una serie de indicadores a los cuales se les asigna un valor que posteriormente se promedia y permite determinar si las revisiones pueden comenzar o no. Estos aspectos a tener en cuenta son:

1. Se presenta la última versión realizada de los documentos.
2. Cantidad de documentos entregados.
3. Se cumplen las reglas para la documentación pactadas en la capacitación inicial.
4. Los documentos entregados se corresponden con la última versión implementada.

Es de suma importancia aclarar que si no se presenta la última versión realizada de los documentos, las pruebas exploratorias se detienen porque se estaría realizando trabajo innecesario. Por tanto, un requisito imprescindible a cumplir es que los proyectos mantengan al día el control de versiones.

Para los restantes indicadores la evaluación se realizará de la siguiente forma:

Si el por ciento de documentos que cumplen con el aspecto a evaluar es menor que el 50% del total de los documentos, no se realizarán las revisiones.

De lo contrario, se ubica el valor obtenido en el siguiente rango para obtener un número entre uno y cinco que permita dar una evaluación cuantitativa del estado inicial de la documentación. Mientras más elevado sea el valor alcanzado, menor será el número de no conformidades identificadas:

- 50%-60% el indicador tendrá peso 1.
- 60%-70% el indicador tendrá peso 2.

CAPÍTULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

- 70%-80% el indicador tendrá peso 3.
- 80%-90% el indicador tendrá peso 4.
- 90%-100% el indicador tendrá peso 5.

Una vez obtenidos los resultados de todos los indicadores, se procederá a calcular un resultado promedio (P) para obtener la evaluación final que determinará si se pueden comenzar o no las revisiones.

Este promedio se calculará de la siguiente manera:

- Se suman los pesos de cada indicador (PI) y el valor obtenido se divide entre la cantidad de indicadores (CI):

$$P = \frac{\sum_{K=0}^n PI}{CI}$$

Imagen 4. Fórmula para calcular el resultado promedio de los pesos alcanzados por los indicadores.

El valor obtenido luego de la realización del cálculo, permitirá evaluar de forma cuantitativa el estado de la documentación, si el resultado obtenido es uno o dos, significa que no se entregaron cantidades significativas de documento que ameriten comenzar una revisión, o que tenían serios problemas de actualización o de correctitud por no corresponderse con lo implementado hasta el momento, de manera general están en mal estado.

Si el valor obtenido es tres, significa que los documentos no están del todo mal, pero tampoco se puede comenzar una revisión con ellos porque la cantidad entregada aún es pequeña, o la correspondencia con lo implementado y la actualización aún presentan defectos significativos que pueden entorpecer las futuras revisiones. El estado de la documentación en este caso es regular.

Por último, si el valor obtenido es cuatro o cinco significa que se entregaron todos o casi todos los documentos necesarios, que los mismos están bastante actualizados y que además, la gran mayoría se

corresponde con lo implementado; por lo que se pueden comenzar las revisiones e informar que la calidad de la documentación es buena.

De la ejecución de esta actividad se encargarán dos miembros del equipo de trabajo, en caso de que sea un proyecto pequeño, el trabajo puede ser realizado por un miembro. Al concluir esta tarea, se debe llenar la Planilla de Resultados de las Pruebas Exploratorias (Ver Anexo 1), en la cual quedará estipulado si el proyecto puede someterse a la evaluación de la documentación o no.

2.3 Descripción de las Actividades

Planificación de las Revisiones

El objetivo principal de esta actividad es crear un cronograma de revisión para cada proyecto, en función de su complejidad y la cantidad de documentos que genera. Además obliga a los proyectos a mantener la documentación en regla, ya que serán evaluados periódicamente. Por la importancia de esta actividad en aras de evitar malos entendidos, se considera fundamental dentro del proceso de evaluación.

En este aspecto es necesario destacar que a cada etapa de revisiones se le denominará Iteración. Por lo que la cantidad de iteraciones y la frecuencia de las mismas quedarán definidas dentro del Cronograma de Revisiones.

En esta tarea interviene el Jefe de Proyecto y el Jefe del Laboratorio de Pruebas como miembro de la dirección del Grupo de Calidad, se debe realizar una reunión entre ambos en aras de llegar a acuerdos relacionados con la frecuencia con que se realizarán las revisiones, la preparación y capacitación del equipo que llevará a cabo la evaluación, así como la garantía de total atención por parte del proyecto al equipo que está llevando a cabo la revisión.

El Jefe del Laboratorio de Pruebas presentará una propuesta de la Lista de Chequeo que se empleará para evaluar los documentos, y se valorará si es necesario incluir u omitir algún aspecto en dependencia de la complejidad, alcance del proyecto y de la iteración que se esté realizando, se propondrá también un cronograma de evaluaciones o revisiones que puede ser modificado en el marco de esa reunión hasta garantizar su aceptación por parte del Líder de Proyecto.

CAPÍTULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

El cronograma propuesto debe ser flexible a cambios, puesto que será necesario tener en cuenta las afectaciones provocadas por actividades docentes, extracurriculares o cualquier otra que pueda incidir en el cumplimiento del mismo. Además, el Líder de Proyecto debe comprometerse a garantizar la capacitación del equipo que llevará a cabo la evaluación. Como resultado de esta reunión se determinará la fecha de cumplimiento de los acuerdos tomados.

En esta actividad intervienen los roles de Jefe de Proyecto y Jefe de Laboratorio de Pruebas. Los documentos consultados serán: Horarios Docentes y de Producción, y se generarán el Cronograma de Revisiones y un Acta de Reunión de Inicio de las Revisiones, donde se recogerán los acuerdos tomados en la misma, así como el compromiso del Jefe de Proyecto con la capacitación del equipo de trabajo que hará la evaluación.

Capacitación del Equipo de Trabajo

El objetivo de esta actividad es presentarle al equipo de evaluación el producto que debe revisar, y lograr que conozcan el mismo como cualquier miembro del equipo de desarrollo del proyecto. Mientras más información posean los miembros del equipo de evaluación sobre el proyecto a revisar, los análisis y búsquedas de defectos y errores en los documentos pueden hacerse más profundos y rápidos, por lo que no se le debe restar importancia a esta actividad.

En esta actividad intervienen los roles de Jefe de Proyecto, Jefe de Laboratorio de Pruebas, Diseñador de Pruebas (que estará al frente del equipo de trabajo) e Ingeniero en Pruebas de Sistema. Se consultarán los materiales necesarios para la capacitación, los casos de uso, el sistema de evaluación de los miembros del equipo de trabajo y el reglamento del GCF, además se generará un plan de trabajo para cada rol, y un registro de control de asistencia y evaluación para el equipo de trabajo.

Ejecución de las Revisiones

El objetivo de esta actividad es ejecutar las revisiones según la planificación realizada, y una serie de pasos definidos para la realización de las pruebas especificado para su ejecución. Esta tarea recae

CAPÍTULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

directamente en el equipo de evaluadores, los cuales tienen el deber de cumplir con el cronograma de trabajo previamente definido.

Las revisiones se realizarán a través de la ejecución de una serie de pasos similares a los desarrollados por el proceso de pruebas de sistema del GCF que han sido modificados específicamente para esta actividad, los mismos se explicarán posteriormente. Es importante destacar que las revisiones pueden ser detenidas antes de que concluyan, si se encuentran algunos de los defectos siguientes:

- Incoherencia significativa entre lo plasmado en el documento de requisitos y los documentos que se revisan.
- La cantidad de faltas de ortografía excede la cantidad de páginas del documento.
- A partir de la segunda iteración persisten al menos dos defectos significativos de iteraciones anteriores.
- Se observan los mismos tipos de defectos ya señalados en iteraciones anteriores en otros o en los mismos lugares donde fueron detectados inicialmente.

En caso de que sea necesario abortar las revisiones por alguna de las causas anteriores, se debe llenar la Planilla de Criterios para Detener las Revisiones (Ver Anexo 2). Además es necesario que se fije un plazo para que se corrijan estos defectos antes de la próxima iteración, la cual se iniciará con una prueba exploratoria que comprobará si los problemas que provocaron el cierre de las revisiones fueron erradicados. En caso afirmativo se seguirá con el cronograma pactado en la Planificación de las Revisiones, de lo contrario, se comenzará todo el proceso nuevamente, a fin de realizar la planificación de las iteraciones nuevamente para lograr revisar todos los documentos antes de que comience el Proceso de Pruebas de Sistema que aplica el GCF.

Al finalizar cada iteración, el Diseñador de Pruebas debe revisar y confirmar las no conformidades detectadas a lo largo del proceso de pruebas llenando la Planilla de No Conformidades de las Revisiones (Ver Anexo 3) donde se archivarán aquellas que hayan quedado en espera de solución al concluir las pruebas, así como efectuar una reunión con el equipo de revisión para puntualizar las incidencias de la iteración en general, las recomendaciones necesarias al equipo de desarrollo del proyecto, y hacerle llegar

CAPÍTULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

el acta de esta reunión conjuntamente con la Planilla de No Conformidades de las Revisiones al Jefe de Proyecto, con el cual se reunirá posteriormente para discutir estos documentos.

A partir de la segunda iteración, las revisiones deben comenzar con una prueba exploratoria en aras de garantizar que las no conformidades identificadas en la iteración anterior fueron respondidas y que las soluciones dadas no hayan generado nuevos defectos ni incompatibilidades entre lo documentado y lo implementado.

Cada iteración de revisiones concluye cuando todos los documentos involucrados en la misma han sido revisados. Luego el Diseñador de Pruebas realizará la misma secuencia de acciones en conjunto con el Jefe de Proyecto. A medida que las iteraciones avancen, la cantidad de no conformidades identificadas debe disminuir.

En caso de que en una iteración no se encuentren no conformidades, se pueden cerrar las revisiones temporalmente, hasta que el proyecto realice modificaciones en la aplicación que deban ser documentadas, o en caso de que no se le agregue nada más al producto, hasta que el mismo solicite el servicio de pruebas del GCF, cuando de forma obligatoria se deberá realizar una revisión profunda y detallada para garantizar que las pruebas puedan comenzar con los documentos completos, correctos y actualizados.

En esta actividad intervienen los roles de Jefe de Proyecto, Diseñador de Pruebas e Ingeniero en Sistemas. Se revisarán los Casos de Pruebas, Listas de Chequeo, Historias de Usuario y Manuales de Usuarios, se generarán la Planilla de No Conformidades de las Revisiones y el Acta de la Reunión entre el Jefe del Servicio de Pruebas y los miembros del equipo de evaluación, y se modificará el registro de asistencia y evaluación del equipo de evaluación.

Evaluación de las Revisiones

Las medidas para evaluar la calidad siempre deberían ser absolutas, sin embargo estas medidas absolutas generalmente no se pueden encontrar, por lo que hay que realizar medidas relativas que permitan formar un criterio a cerca del estado en que se encuentra lo que se está midiendo.

CAPÍTULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

En este caso es necesario auxiliarse de herramientas para la medición de la calidad de los productos, como por ejemplo, la métrica para la densidad de defectos. Esta se calcula como el total de los defectos conocidos dividido entre el tamaño de la entidad de software que se mide. $Densidad = \frac{\text{Número de defectos conocidos}}{\text{Tamaño}}$. El número de defectos puede ser el siguiente (32):

- Defectos hasta la fecha desde la creación del módulo
- Defectos encontrados en un programa durante una inspección
- Defectos hasta la fecha desde el envío de una liberación al cliente.

El tamaño es un normalizador que permite las comparaciones entre las diferentes entidades de software que pueden ser módulos, productos, etc. Tamaño suele ser contado, ya sea en líneas de código o en puntos de función.

Como se puede apreciar, esta métrica está dirigida a la medición del software como producto, pero es importante destacar que los documentos son la manifestación tangible del software, además, también son considerados un producto. Por esta razón se ha realizado una adaptación de la métrica para emplearla en la evaluación de las revisiones en general.

El tamaño va a ser la cantidad de páginas revisadas y el número de defectos será entonces:

- La cantidad de no conformidades detectadas relacionadas con la actualización del documento.
- La cantidad de no conformidades detectadas relacionadas con la correctitud del documento.
- La cantidad de no conformidades detectadas relacionadas con la completitud del documento.
- Total de no conformidades detectadas en la revisión del documento.

Primeramente se calcula la densidad del total de no conformidades, si el valor obtenido es mayor que tres, significa que hay aproximadamente tres defectos por página, por lo que la evaluación que obtendrá el proyecto en esa revisión será insatisfactoria.

En caso contrario, se procede a calcular la evaluación de la revisión a través del siguiente grupo de pasos:

1. Para cada documento:

CAPÍTULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

- a. Del total de no conformidades identificadas se calcula el porcentaje que representan las relacionadas con la correctitud, completitud y actualización por separado.
- b. Se ubica el porcentaje calculado en el siguiente rango:
 - 80%-100% peso 1.
 - 60%-80% peso 2.
 - 40%-60% peso 3.
 - 20%-40% peso 4.
 - 0%-20% peso 5.

En este caso obtener peso uno o dos significa que prácticamente la totalidad de no conformidades identificadas van a estar relacionadas con la misma temática (completitud, correctitud o actualización), por lo que la temática en cuestión va a tener una presencia fuerte dentro del documento. Tener peso tres quiere decir que aproximadamente la mitad de las no conformidades identificadas van a ser del mismo tema, así que se presencia dentro del documento es media. Por último tener peso cuatro o cinco, significa que menos de la mitad de las no conformidades identificadas van a tener la misma temática, así que se manifiesta débilmente dentro del documento.

2. Se suma la puntuación obtenida por la actualización, correctitud y completitud en cada documento y se divide entre la cantidad de documentos revisados.
3. Se calcula el porcentaje que representan la cantidad de documentos que presentan el tipo de defecto anteriormente mencionado del total de documentos y se le asigna un peso según el siguiente rango:
 - 50%-60% la subcaracterística tendrá peso 1.
 - 60%-70% la subcaracterística tendrá peso 2.
 - 70%-80% la subcaracterística tendrá peso 3.
 - 80%-90% la subcaracterística tendrá peso 4.
 - 90%-100% la subcaracterística tendrá peso 5.

4. Por último se obtendrá un valor a través de la siguiente fórmula, el cual será la evaluación final de la revisión:

$$ER = \frac{\sum_{k=0}^n (I * P)}{\sum_{k=0}^n P}$$

Imagen 5. Fórmula para la obtención de la evaluación de la revisión.

Donde ER es la evaluación de la revisión, I es el valor obtenido luego de la aplicación del paso número 2 y P es el valor obtenido luego de la aplicación del paso número 3. El valor de ER es un número entre 1 y 5, siendo 5 el mejor resultado y 1 el peor, lo cual permite darle una evaluación cuantitativa a la revisión. Mientras más alto sea el resultado de la evaluación se podrá decir que el número de no conformidades relacionadas con la actualización, completitud y correctitud que se encontrarán durante la aplicación del Proceso de Pruebas de Sistema del GCF será menor.

Si el resultado de la evaluación es menor que cuatro, es necesario realizar otra iteración de pruebas a los documentos, puesto que aún presentarán cantidades significativas de defectos que pueden influir negativamente en los resultados del Proceso de Pruebas del GCF.

Esta actividad es desarrollada por todos los miembros del equipo de trabajo y dirigida por el Responsable del Servicio de Pruebas. Además, al concluir la misma se le da la evaluación final de la revisión a cada miembro del equipo. Se modifica el registro de asistencia y evaluación del equipo evaluador.

Cierre de las Revisiones

El objetivo de esta actividad es dar por concluido el proceso de revisiones, y proporcionar una valoración final del estado de la documentación al Jefe de Proyecto. Cuando se concluyen las revisiones, se debe realizar una reunión final donde se determina si la documentación del proyecto cuenta con la calidad necesaria para que el mismo se pueda enfrentar al proceso de pruebas o no.

CAPÍTULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

Durante la ejecución de esta actividad se valora si todas las no conformidades fueron respondidas, de ser así, el proyecto está en condiciones de realizar una solicitud de pruebas al GCF, en caso negativo, se debe hacer un análisis de las no conformidades que no fueron respondidas y a partir del mismo valorar si las mismas inciden de forma negativa en el desarrollo del proceso de pruebas establecido por el GCF.

Los roles que intervienen en esta tarea son el Jefe de Proyecto y Diseñador de Pruebas. Se consulta nuevamente el cronograma definido durante la ejecución de la primera actividad para comprobar que el proceso se desarrolló según lo pactado, y se genera un acta de la reunión que se desarrolla, la cual debe recoger las no conformidades que no fueron respondidas, el nivel de cumplimiento del cronograma pactado, y si el proyecto puede solicitar el servicio de pruebas o no.

Las actividades anteriormente descritas aparecen reflejadas en la siguiente imagen:



Imagen 6. Proceso de evaluación de la documentación para el servicio de pruebas funcionales del GCF.

2.4 Descripción de los pasos de pruebas a seguir para las revisiones

Cuando se trazan un conjunto de actividades a realizar durante la evaluación de documentos, es de referencia obligatoria el término proceso de pruebas. Dada la necesidad de obtener un producto libre de fallas, y conociéndose que esto depende en gran medida de la cantidad de fallas que tengan los documentos del mismo, se recurrió a la elaboración y aplicación de un proceso de pruebas a la documentación de los proyectos productivos de FORTES, que describiera y registrara el trabajo a desempeñar por un equipo de trabajo que a partir de ese momento se dedicaría de forma exclusiva a esta actividad y que sirva de apoyo al proceso de evaluación de los documentos.

El mismo se conformó siguiendo una serie de pautas definidas en el proceso de pruebas de sistema del GCF, las cuales permiten agilizar el trabajo. El objetivo es llevar un adecuado control de los documentos a probar, así como los probados y los resultados obtenidos en la presente y anteriores iteraciones.

Dicho proceso integra un conjunto de actividades definidas para cada una de las etapas propuestas. Comienza con la planificación de las pruebas, donde se definen la estrategia, alcance y objetivos de las pruebas a aplicar, le sigue el diseño de la misma, abarcando todo lo relacionado con el procedimiento a seguir y con la herramienta de recolección de la información que se utilizará para contabilizar los resultados obtenidos, y por último se procede a probar y evaluar, para un posterior análisis de la calidad de las mismas a partir de los resultados obtenidos.

Se necesita por parte de los líderes de los proyectos de FORTES, la realización a tiempo de los documentos, dígame: casos de prueba, listas de chequeo, manuales de usuario, historias de usuario y especificaciones de casos de uso, en caso no estar incluidos en la especificación de casos de uso, un documento que contenga los requerimientos y a que caso de uso corresponde.

Luego de aplicadas las pruebas se genera el documento de defectos y no conformidades, como herramienta de recolección de los resultados obtenidos. Contempla una descripción detallada de todos los defectos y errores detectados, así como sugerencias realizadas al equipo de desarrollo. Los resultados reales (satisfactorios o insatisfactorios), determinarán si el GCF acepta o no las solicitudes de pruebas realizadas por los proyectos de FORTES.

CAPÍTULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

La estrategia seguida concibe probar los documentos de forma periódica, considerando cada una como una iteración dependiente de la anterior. Para ello se cuenta con una lista de chequeo que incluye los aspectos de mayor importancia a revisar, y cómo deben probarse los mismos, además, se hace uso de una plantilla que recoge los defectos o errores que pueden incidir con mayor fuerza en la futura utilización de los documentos.

A partir de la propuesta realizada sobre las pruebas a aplicar a la documentación, se describen brevemente las etapas que conforman el proceso:

- **Planificar Prueba:** Consiste en realizar una planificación detallada del día y horario en que se probará cada documento. Los principales responsables de esta actividad son el Jefe de Proyecto y el Responsable del Servicio de Pruebas, que hará las veces del líder del equipo de pruebas. Se consulta el cronograma definido en la planificación del proceso de revisiones y se genera un plan de trabajo para esa iteración.
- **Diseñar Prueba:** El propósito general de esta actividad es definir y diseñar los requisitos de datos para ejecutar las pruebas y los métodos para registrar los resultados. El peso de esta tarea recae en el Diseñador de Pruebas y consiste, básicamente, en realizar las modificaciones pertinentes a la lista de chequeo pactada en la fase de planificación del proceso de evaluación, para que se adapte a los objetivos específicos de la iteración. En esta actividad se consulta y modifica la Lista de Chequeo.
- **Ejecutar Pruebas:** Abarca la ejecución de las pruebas anteriormente diseñadas con el fin de evaluar la completitud, correctitud y actualización de los documentos, y en qué medida estos aspectos se han cumplido o no. Comprende además la documentación de los errores que se van detectando para darles seguimiento hasta garantizar la total corrección de los mismos. En esta actividad intervienen todos los miembros del equipo de evaluación. Se obtiene una Planilla de No Conformidades (vea Anexo 4), donde se recogen los errores y defectos detectados durante las sesiones de trabajo.

- **Evaluar Pruebas:** En esta etapa se analizan los resultados e incidencias diarias de las pruebas a partir de los diferentes aspectos que puedan haberlos ocasionado, así como la repercusión que tienen en la calidad del documento en general. Se define además si se pueden continuar las pruebas o si los errores detectados impiden el desarrollo de las mismas.

Esta tarea la desarrollan el Jefe de Proyecto y el Responsable del Servicio de Pruebas. Además se genera un documento que recoge la evaluación diaria del trabajo, la cual tributa directamente a la evaluación general de la iteración. El éxito de una evaluación está en no depender de la memoria, o sea, en obtener los datos cuando se están generando, no después de concluidas las pruebas. Por esta razón es vital para esta etapa que los evaluadores llenen la Planilla de No Conformidades en el momento en que las detectan y no al finalizar la jornada laboral.

2.5 Organización del Equipo de Trabajo

El equipo encargado de llevar a cabo las pruebas está compuesto por los siguientes roles:

- Diseñador de Pruebas (planificar, diseñar y evaluar)
- Ingeniero en Pruebas de Sistema (ejecutar pruebas)

La cantidad de personas que conformen el mismo puede variar en dependencia de la magnitud del proyecto que se esté revisando, mientras más documentos tenga el proyecto, más personas se necesitarán para realizar las pruebas, aunque se recomienda que el equipo sea relativamente pequeño.

El proceso de pruebas que se describe sigue el comportamiento de una prueba exploratoria, donde el éxito de la misma recae, principalmente, en las habilidades de los evaluadores que la realizan, siendo indispensable para su ejecución que el equipo de pruebas conozca a fondo el funcionamiento del producto, a quién va destinado y lo que se espera alcanzar con él, o sea, que durante el tiempo que duren las pruebas, se integren como miembros del equipo de desarrollo. Por esto se hace necesario que el equipo de trabajo esté conformado por las personas que más experiencia tengan dentro del GCF, ya que deben poseer conocimientos profundos sobre la realización de cualquiera de los documentos a probar, lo cual facilita que las pruebas se realicen más rápido y con mejor calidad.

Es importante señalar que los roles de Diseñador de Pruebas y de Ingeniero en Pruebas, pueden ser desempeñados por la misma persona, no es necesario separarlos ya que en este tipo de pruebas el trabajo de ambos es similar y se resume, esencialmente, en la ejecución de las pruebas. Por su parte el Responsable de Servicio de Pruebas hará la función de líder del grupo y se encargará de almacenar y organizar las no conformidades emitidas, y en conjunto con los miembros del proyecto que se está revisando determinar si las mismas proceden o no.

2.6 Conclusiones del Capítulo

Se presentó un proceso capaz de integrarse con los demás procesos que se llevan a cabo en el CGF. De manera general se establecieron para las actividades o tareas del proceso: entradas, salidas y responsables, así como tareas de apoyo para garantizar el éxito de las mismas. Se definieron los roles del proceso basados en los existentes dentro del GCF y en las experiencias acumuladas por la autora y tutora de la investigación, en función de mantener una única línea de trabajo dentro del grupo, así como evitar la confusión de los estudiantes y personas implicadas en el mismo.

Para la aplicación exitosa de la propuesta de solución hecha en la presente investigación se definieron herramientas documentadas de vital importancia para el proceso, las cuales garantizan un apoyo fundamental para la aplicación del mismo. De manera general esta propuesta se logró enfocar a resolver la problemática que inicialmente existía.

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN PROCESO DE EVALUACIÓN DE LA DOCUMENTACIÓN PARA EL SERVICIO DE PRUEBAS FUNCIONALES A NIVEL DE SISTEMA QUE OFRECE EL GRUPO DE CALIDAD DE FORTES.

3.1 Introducción

Es necesario asegurar que el proceso sea diseñado, desde sus inicios, enfocado al objetivo deseado y con la capacidad de mejorar en el tiempo. Se estudiaron metodologías que permitan la construcción y aseguren la mejora continua del mismo. Además, se mostrarán los resultados obtenidos luego de la implantación de la solución propuesta en los casos de estudios seleccionados.

3.2 Aplicación de la solución propuesta

En el proceso de introducción del Six Sigma, uno de los conceptos que más se aplica, son una serie de pasos conocidos por sus siglas DMAIC, con lo cual se busca establecer la fuente u origen de la variación. La D, significa Definir, la M es Medir, la A es Analizar, la I corresponde a la palabra en inglés Improve, que equivale a Mejorar y la C es Controlar. Se puede considerar entonces como una modificación para la mejora continua de Planear, Hacer, Verificar y Actuar (PDCA).

Siguiendo las normas planteadas en el proceso de evaluación desarrollado, las cuales especifican que el equipo de trabajo debe ser pequeño y formado por las personas que más experiencia tengan en la ejecución de las pruebas, se seleccionó como una población confiable para obtener la información un grupo de cinco estudiantes que cursan el 5to año con amplia experiencia en la realización de pruebas y que conocen a fondo las características del GCF.

3.2.1 Casos de estudio

Para el presente estudio fueron definidos teniendo en cuenta su componente software y la metodología empleada para su realización seis trabajos de diploma, los cuales no han comenzado la fase de pruebas correspondiente. De ellos, tres se desarrollaron empleando metodologías ágiles, específicamente XP, y los restantes emplearon RUP.

Los equipos de desarrollo de las muestras están conformados por un máximo de dos personas, los cuales se desempeñaron en todos los roles definidos por la metodología empleada, lo cual facilitó la aplicación del proceso de evaluación, ya que los desarrolladores tenían un profundo conocimiento del producto así como de su funcionamiento.

Durante el período se desarrollaron los roles de Diseñador de Pruebas e Ingeniero en Pruebas de Sistema. El equipamiento básico estuvo conformado por cinco laptops de uso personal de los miembros del equipo. Se dedicó una de las laptop como repositorio de la información generada durante el proceso.

3.3 Resultados de la aplicación

En este epígrafe se analizan las variables de control de Six Sigma, desde la óptica de la introducción piloto del resultado de la investigación, así como los indicadores y parámetros de la investigación, con el objetivo de demostrar la idea a defender inicial.

Definir

El proceso de evaluación que en esta investigación se propone, responde a la necesidad de mejorar la calidad de la documentación que sirve de entrada al proceso de pruebas que desarrolla el GCF, garantizando una mejor y más rápida revisión de los productos y mejorando el estado de ánimo de los probadores puesto que no será necesario realizar un doble esfuerzo por causa de la documentación en mal estado.

Por lo anteriormente expuesto se puede decir que el proceso de evaluación propuesto por esta investigación es un proceso estratégico para el GCF, porque del mismo puede depender, en gran medida un aumento en la calidad con que se realizan las pruebas.

Dentro de las actividades definidas para garantizar una ejecución satisfactoria del proceso anteriormente mencionado, la que cuenta con mayor prioridad para la realización de mejoras es la Ejecución de las Pruebas Exploratorias, puesto que los parámetros que se evalúan en la misma, deben actualizarse en función del producto que se revise, además deben ser flexibles a los cambios que puedan surgir en los

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

estándares o formato de los documentos que se revisan, o de las metodologías que se empleen en la universidad.

Medir

El cliente potencial y a quien va dirigido el resultado de la presente investigación, es el GCF, sin embargo, por sus características, puede ser utilizado por cualquier grupo de calidad dentro de la UCI. Este proceso se creó para suplir la necesidad del GCF de controlar la calidad de la documentación que llega como entrada al proceso de pruebas, de esta forma se puede reducir el número de no conformidades que no proceden para los líderes de los proyectos que se revisan, así como disminuir los atrasos que la documentación incorrecta, incompleta o desactualizada ocasiona.

La propuesta de solución que se plantea en esta investigación, consta de seis actividades fundamentales, encaminadas a alcanzar la satisfacción del cliente y la resolución del problema que al inicio de la misma se plantea. El orden de realización propuesto para cada actividad debe respetarse porque de lo contrario el proceso pierde su funcionalidad y objetivo puesto que la realización de una actividad depende del éxito en la ejecución de la anterior.

Los aspectos que se tuvieron en cuenta para medir la efectividad del proceso están dados por el grado de actualización, completitud y correctitud que tengan los documentos luego de la aplicación del mismo. Mientras más elevado sea el valor numérico obtenido, más efectivo será el proceso.

Los parámetros anteriormente mencionados responden a las necesidades críticas del cliente de mejorar el estado de la documentación, antes de que la misma llegue al proceso de pruebas. Para ello se realizó un estudio previo al desarrollo de esta investigación, el cual arrojó que la no completitud, la no correctitud y la desactualización de los documentos que sirven como entrada al proceso de pruebas desarrollado por el GCF, son las principales causas que sustentan la situación problemática que se desea erradicar con la solución que se propone.

Analizar

Teniendo en cuenta los parámetros que se determinaron como necesidades críticas del cliente el proceso cuenta con listas de chequeo para evaluar los mismos en cada ejecución. Estas listas de chequeo deben modificarse dependiendo del tipo de documento que se revise, o de la naturaleza del proyecto al que los mismos pertenecen. El Anexo 5 muestra la lista de chequeo que se empleó para evaluar el comportamiento de los indicadores en una de las muestras seleccionadas.

Luego de analizar los resultados obtenidos después de la aplicación del proceso, se determinaron como objetivos de mejora las vías para calcular la evaluación final, puesto que se basan en una serie de fórmulas matemáticas que aportan complejidad a la actividad y pueden causar confusión en su utilización.

Las fuentes de variación del proceso radican en cuatro aspectos fundamentales:

- La metodología empleada para el desarrollo del proyecto a evaluar.
- Formato empleado en la elaboración de los documentos que se utilizan como entrada para el proceso de pruebas del GCF.
- Estándares y normas para la calidad establecidos por la UCI
- El proceso de pruebas funcionales a nivel de sistema que se lleva a cabo en el GCF.

Lamentablemente estas fuentes de variación, no pueden ser controladas desde el proceso, puesto que resulta difícil prevenir la naturaleza del cambio y el estado final de las variables luego del mismo. Siendo esta la razón fundamental por la cual se decidió aplicar la metodología Six Sigma para la mejora del proceso.

Mejorar

Actualmente, las fuentes de variación identificadas no son un peligro eminente capaz de colapsar el proceso de evaluación propuesto en esta investigación, además, no se conocen las posibles raíces o naturalezas de estos cambios, pues la UCI se encuentra en constante renovación y mejora.

Es necesario tener en cuenta que las listas de chequeo son el elemento más flexible a cambios dentro del proceso definido, por lo que pueden representar la solución a varias de las fuentes de variación

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

identificadas, las mismas se definen en la segunda actividad del proceso, y en acuerdo con personas que conocen a fondo su proyecto y las necesidades del mismo.

Controlar

Luego de la aplicación de la propuesta de solución hecha en esta investigación, no se puso de manifiesto ninguna de las posibles fuentes de variación identificadas, por lo que no fue necesario realizarle modificaciones.

Después de analizar los resultados obtenidos luego de la aplicación de los pasos definidos por Six Sigma para la construcción y mejora del proceso se puede concluir que se ha desarrollado un proceso de evaluación con una visión muy clara de hacia donde se quería llegar dejando explícito el enfoque al cliente y satisfaciendo su deseo de servirse de una herramienta que en pocos pasos y de sencilla aplicación, garantizara la calidad de la documentación que sirve como entrada al proceso de pruebas del GCF.

3.4 Análisis de los casos de estudio

Para realizar un análisis más exacto y minucioso a cerca de los casos de estudio que se tomaron como muestra para la aplicación de la propuesta de solución que brinda la presente investigación, y respetando el anonimato de las mismas por cuestiones éticas, se decide nombrarlas a través de variables para lograr una mejor identificación e independencia como se evidencia a continuación.

| Muestra | Metodología Empleada | Cantidad de HU | Cantidad de CP | Cantidad de MU | Cantidad de LC | Temática |
|----------------|-----------------------------|-----------------------|-----------------------|-----------------------|-----------------------|--------------------------------|
| A | XP | 30 | 30 | - | - | Portal web dedicado al deporte |
| B | ScrumXP | 19 | 19 | - | - | Catálogo online de servicios |
| C | XP | 4 | 4 | - | - | Compilador |
| D | RUP | - | 17 | 2 | - | Herramienta |
| E | RUP | - | 21 | - | 4 | - |

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

| | | | | | | |
|---|-----|---|----|---|---|---|
| F | RUP | - | 38 | - | - | - |
|---|-----|---|----|---|---|---|

Tabla 1. Desglose de los casos de estudio

En el caso de la muestra A, se decidió aplicar el proceso empleando como documentación para el desarrollo de las pruebas las historias de usuario. En ocasiones resulta de mayor comprensión para el probador el uso de las mismas en lugar de los casos de prueba. Esta muestra permitió refinar el proceso y ejecutarlo en varias ocasiones antes de su validación.

Para la muestra B, se decidió aplicar el proceso con ambos tipos de documentos (historias de usuario y casos de prueba) para realizar una comparación y determinar cual resulta más eficiente a la hora de detectar no conformidades, y donde se comenten más errores durante la realización.

La muestra C se probó usando solamente los casos de prueba, y de esta forma determinar la eficiencia del proceso en caso de que las historias de usuario no estuvieran elaboradas correctamente, o fueran inexistentes.

El caso D se tomó como muestra para comprobar si la propuesta de solución se puede emplear para probar las correspondencias entre lo implementado y lo estipulado en el manual que recibirá finalmente el usuario.

E es una muestra atípica, puesto que ya había sido sometida a procesos de pruebas anteriormente, con resultados poco satisfactorios, además, cuenta con listas de chequeo de usabilidad para ser empleadas específicamente en el proceso de pruebas del GCF, la temática de esta muestra se decidió mantener en el anonimato porque así lo solicitaron los desarrolladores de la misma.

Por último se tomó la muestra F, que por su complejidad y similitud con la mayoría de los proyectos que se prueban dentro del GCF se decidió dejar para el final y así realizar un proceso de revisión y evaluación más minucioso y detallado. Al igual que la muestra E, se decidió mantener en el anonimato la temática de la misma por solicitud de sus desarrolladores.

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

3.4.1 Análisis comparativo con la situación anterior

Antes de aplicar la propuesta de solución, el equipo de trabajo decidió someter al proceso de pruebas del GCF todas las muestras seleccionadas y se registraron todos los resultados obtenidos para futuras comparaciones. Posteriormente se aplicó el proceso de evaluación desarrollado a las mismas muestras, y de igual forma se archivaron los resultados obtenidos con el mismo objetivo anteriormente mencionado.

Para el inicio de las evaluaciones de los resultados, se ubicaron ambos registros obtenidos antes y después de la aplicación del proceso de evaluación de la documentación. A continuación se hace un breve resumen de las principales temáticas en torno a las que giraban las no conformidades identificadas.

- Los documentos no se corresponden con lo que está implementado.
- Lo que se plantea en la documentación no se corresponde con lo que plantea al cliente en las historias de usuario y documentos de requisitos.
- Faltan datos imprescindibles para poder probar la aplicación.

Luego de llevar a cabo el proceso en las muestras seleccionadas, las temáticas sobre las que giraban las no conformidades eran de menor importancia, por ejemplo, el formato de los documentos, o el tamaño de la letra en la aplicación. A continuación se muestra una tabla comparativa donde se evidencian los resultados antes y después de la aplicación de la propuesta de solución para cada muestra:

| MUESTRA | ANTES | | | | | DESPUÉS | | | | |
|--------------|------------|-----------|-----------|-----------|-----------|-----------|----------|-----------|----------|-----------|
| | NC | NC Act | NC Corr | NC Comp | Otras NC | NC | NC Act | NC Corr | NC Comp | Otras NC |
| A | 24 | 4 | 8 | 5 | 7 | 12 | 1 | 4 | 2 | 5 |
| B | 22 | 5 | 10 | 3 | 4 | 9 | 3 | 2 | 1 | 3 |
| C | 10 | 2 | 6 | 1 | 1 | 2 | 0 | 0 | 0 | 2 |
| D | 15 | 1 | 3 | 2 | 9 | 8 | 0 | 1 | 1 | 6 |
| E | 11 | 2 | 3 | 2 | 6 | 6 | 0 | 1 | 1 | 4 |
| F | 50 | 10 | 20 | 8 | 12 | 22 | 5 | 8 | 4 | 5 |
| TOTAL | 132 | 24 | 50 | 21 | 39 | 59 | 9 | 16 | 9 | 25 |

Tabla 2. Comparación entre el antes y el después de la aplicación de la propuesta de solución.

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

El siguiente gráfico evidencia la disminución de las no conformidades identificadas, y permite ideas más claras de los resultados obtenidos:

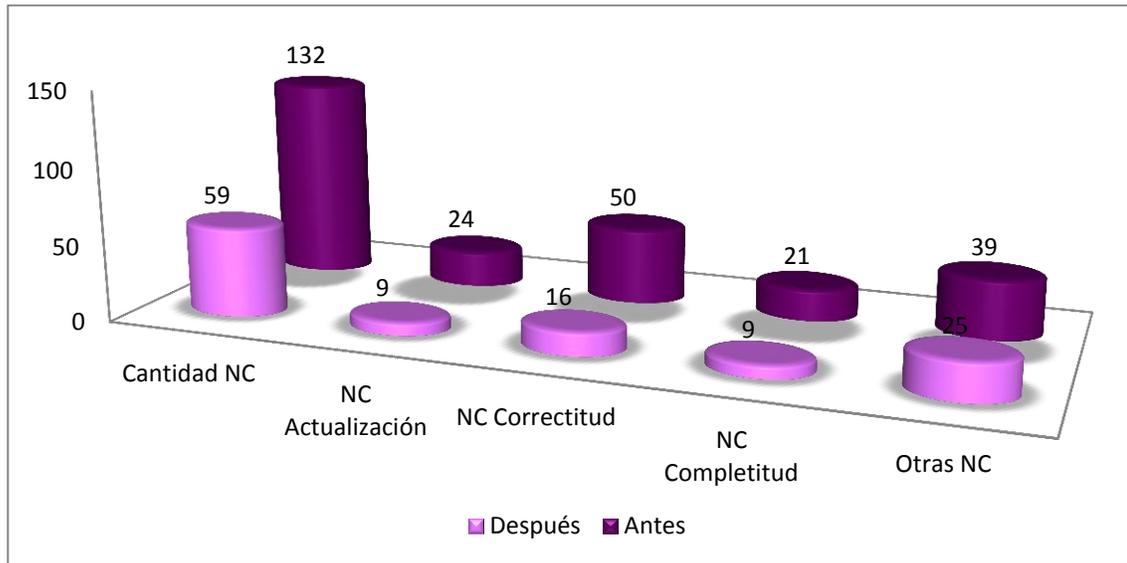


Imagen 7. Gráfico de resultados comparativos de la aplicación de la propuesta de solución.

Luego de analizar los resultados obtenidos se puede decir que los proyectos desarrollados con metodologías ágiles generan un porcentaje menor de no conformidades que los que emplean RUP, esto es consecuencia de las características propias de ambas metodologías, teniendo las metodologías ágiles una ventaja significativa puesto que es más fácil elaborar los documentos que genera en comparación con los de la misma finalidad generados por RUP.

Por lo anteriormente expuesto se hace necesario resaltar que la aplicación del proceso que aquí se propone, resulta de vital importancia para los proyectos de gran envergadura, puesto que las cifras demuestran una reducción entre un 44% y un 54% en la cantidad de no conformidades identificadas relacionadas con la completitud, correctitud y actualización, luego de la puesta en práctica de la propuesta de solución que en esta investigación se hace. Los resultados obtenidos indican que se contribuyó a la mejora de la calidad de la documentación de entrada al Proceso de Pruebas de Sistema del GCF y se espera que las problemáticas que dieron pie a la propuesta de solución queden erradicadas.

3.5 Conclusiones del capítulo

Después de evaluar las etapas definidas por Six Sigma para la construcción y mejora del proceso, así como los resultados obtenidos luego de la aplicación del mismo, se puede concluir que se ha desarrollado un proceso enfocado al cliente. De manera general, la metodología de diseño utilizada garantiza la satisfacción de las necesidades de los clientes de la solución propuesta. Algunos elementos como los pasos la ejecución de las pruebas y el número de actividades, deben ampliarse o trabajarse en las siguientes iteraciones de mejora.

La prueba se realizó para un conjunto de muestras de ensayo y se midieron los parámetros fundamentales del proceso siendo satisfactorios los resultados en prácticamente todas las muestras. A modo de síntesis la solución propuesta coloca al GCF en una mejor posición para el incremento de la calidad del proceso de pruebas funcionales a nivel de sistema que actualmente se desarrolla.

CONCLUSIONES GENERALES

Como conclusiones generales de la investigación se muestran alcanzados los objetivos propuestos de manera satisfactoria. Se constató la existencia en el mundo de una problemática alrededor de la calidad de software, que conlleva al análisis de los procesos de pruebas de software, además se evidenció el esfuerzo existente para revertir la situación con la creación de espacios, estructuras y organizaciones entre otras vías para lograrlo. Se obtuvieron además datos e información relevante de los procesos de pruebas de software que se desarrollan en la UCI, analizando las fortalezas y debilidades que pudieran influir en el resultado final de esta investigación.

Se presentó un proceso capaz de integrarse con los demás procesos que se llevan a cabo en el CGF. De manera general se establecieron para las actividades o tareas del proceso: entradas, salidas y responsables. Lo anteriormente expuesto indica que se elaboró un proceso ordenado de evaluación preventiva a la documentación que sirve como entrada al proceso de pruebas, ajustado al GCF, insertado en un entorno docente-productivo. Se definieron los roles del proceso basados en los existentes dentro del GCF y en las experiencias acumuladas por la autora y tutora de la investigación, en función de mantener una única línea de trabajo dentro del grupo, así como evitar la confusión de los estudiantes y personas implicadas en el mismo. Esta propuesta se logró enfocar a resolver la problemática que inicialmente existía.

De manera general, la metodología de diseño utilizada garantiza la satisfacción de las necesidades de los clientes de la solución propuesta, aunque algunos elementos el número de actividades, deben ampliarse o trabajarse en las siguientes iteraciones de mejora. La prueba se realizó para un conjunto de muestras tomadas como ensayo y se midieron los parámetros fundamentales del proceso siendo satisfactorios los resultados en prácticamente todas las muestras. Esto se logra a partir de aplicar parcialmente el proceso definido. Con la aplicación y maduración del proceso se estima mejore la calidad de la documentación que sirve de entrada al proceso de pruebas funcionales a nivel de sistema desarrollado por el GCF, colocándolo en una posición satisfactoria para la mejora de la calidad del proceso de pruebas funcionales a nivel de sistema que actualmente se desarrolla.

RECOMENDACIONES

- Extender el análisis a los restantes proyectos de la universidad.
- Implantar totalmente el proceso involucrando a los profesores y principales responsables de los proyectos productivos que se desarrollan en la facultad.
- Adecuar la solución propuesta y extenderla a otras instancias como mecanismo de control de la calidad. Dentro de las que se pudieran encontrar los grupos de calidad de las restantes facultades, CALISOFT, empresas radicadas en la UCI, Complejo UCI-FAR y los polos productivos.
- Hacer una segunda iteración, después de 6 meses de implantado, con la herramienta Six Sigma como modo de mejora de procesos. Elementos como los pasos para el desarrollo de las pruebas y el número de actividades, deben ampliarse o trabajarse en las siguientes iteraciones de mejora.
- Incorporar este conocimiento de pruebas de software en el programa docente educativo de la asignatura Práctica Profesional desde segundo año de la carrera, para formar tempranamente estudiantes con cultura de pruebas y las herramientas necesarias para su desempeño como probadores, que sean capaces de desarrollar el proceso propuesto por la siguiente investigación de forma satisfactoria.
- Desarrollar una herramienta web que pueda adicionarse al portal del GCF, con la finalidad de automatizar la actividad de evaluación de las revisiones, por la complejidad y difícil comprensión de los pasos que engloba.

TRABAJOS CITADOS

1. **Gálvez George, Greisy.** *Modelo de Proceso del Servicio de Pruebas de Software a nivel de sistema en FORTES.*
3. **Pressman, Roger S.** *Ingeniería del Software. Un enfoque práctico.* España : Mc Graw-Hill/ Interamericana de España, 2002.
4. **Standard, Sftware Engineering- IEEE.** *Glossary of Software Engineering Terminology.* s.l. : Institute of Electrical and Electronics Engineers, 1990.
5. **Pes, Carlos Javier.** Minidiccionario Informático. [En línea] [Citado el: 14 de enero de 2011.] http://www.carlospes.com/minidiccionario/calidad_del_software.php.
6. **Beizer, Boris.** *Software Testing Techniques.* s.l. : International Thomson Computer Press, 1990.
7. **Española, Real Academia de la Lengua.** *Diccionario de la Real Academia de la Lengua Española. Vigésimo Segunda Edición.* s.l. : ESPASA CALPE, 2003.
8. **Juran, J M, Gryna, Frank M y Bingham, R S.** *Manual de Control de la Calidad.* s.l. : Reverté, 1987.
9. **Reales Mateo, Pedro.** *Calidad de Casos de Prueba.*
11. **Canós, José H., Letelier, Patricio y Penadés, María Carmen.** *Métodologías Ágiles en el Desarrollo de Software.* s.l. : Universidad Politécnica de Valencia.
12. Definición de... [En línea] [Citado el: 12 de enero de 2011.] <http://definicion.de/manual-de-usuario/>.
13. **Clements, Paul, Kazman, Rick y Klein, Mark.** *Evaluating Software Architecture: Methods and Case Studies.* s.l. : The SEI Series in Software Engineering, 2002.
14. **Fenton, Norman y Pfleeger, Shari Lawrence.** *Software metrics: a rigorous and practical approach.* s.l. : International Thomson Computer Press, 1996.
32. **Centro de Excelencia para el Desarrollo de Proyectos Tecnológicos.** *Métricas de defectos.* La Habana : s.n., 2009.

BIBLIOGRAFÍA

1. **Gálvez George, Greisy.** *Modelo de Proceso del Servicio de Pruebas de Software a nivel de sistema en FORTES.*
2. **Feigenbaum, Armand V.** *Control total de la caldiad.* 1951.
3. **Pressman, Roger S.** *Ingeniería del Software. Un enfoque práctico.* España : Mc Graw-Hill/ Interamericana de España, 2002.
4. **Standard, Sftware Engineering- IEEE.** *Glossary of Software Engineering Terminology.* s.l. : Institute of Electrical and Electronics Engineers, 1990.
5. **Pes, Carlos Javier.** Minidiccionario Informático. [En línea] [Citado el: 14 de enero de 2011.] http://www.carlospes.com/minidiccionario/calidad_del_software.php.
6. **Beizer, Boris.** *Software Testing Techniques.* s.l. : International Thomson Computer Press, 1990.
7. **Española, Real Academia de la Lengua.** *Diccionario de la Real Academia de la Lengua Española. Vigésimo Segunda Edición.* s.l. : ESPASA CALPE, 2003.
8. **Juran, J M, Gryna, Frank M y Bingham, R S.** *Manual de Control de la Calidad.* s.l. : Reverté, 1987.
9. **Reales Mateo, Pedro.** *Calidad de Casos de Prueba.*
10. **Méndez, Edumilis, Pérez, María y Mendoza, Luis E.** *Aplicación de un método para especificar Listas de Chequeo. Talleres de Ingeniería de Software y Bases de Datos.* Caracas : Laboratorio de Investigación en Sistemas de Información. Universidad Simón Bolívar, 2007. Vol. 1, 4.
11. **Canós, José H., Letelier, Patricio y Penadés, María Carmen.** *Métodologías Ágiles en el Desarrollo de Software.* s.l. : Universidad Politécnica de Valencia.
12. Definición de... [En línea] [Citado el: 12 de enero de 2011.] <http://definicion.de/manual-de-usuario/>.
13. **Clements, Paul, Kazman, Rick y Klein, Mark.** *Evaluating Software Architecture: Methods and Case Studies.* s.l. : The SEI Series in Software Engineering, 2002.
14. **Fenton, Norman y Pfleeger, Shari Lawrence.** *Software metrics: a rigorous and practical approach.* s.l. : International Thomson Computer Press, 1996.
15. **Schulmeyer, Gordon y McManus, James.** *Handbook of software quality assurance.* s.l. : Prentice Hall, 1998.
16. **Universidad de Guadalajara.** *Métricas, Estimación y Planificación en Proyectos de Software.* Guadalajara : s.n.
17. **Calero Muñoz, Coral.** *Métricas de Software: Conceptos básicos, definición y formalización.* La Mancha : s.n., 2006.
18. **Hall, W. E. y Zweben, S. H.** *The Cloze Procedure and Software Comprehensibility Measurement.* s.l. : IEEE Transactions on Software Engineering, 1986.
19. **Lehner, F.** *Quality control in software documentation: Measurement of text comprehensibility.* 1993. págs. 133-146.
20. **Jonhson, S.** *Control for Hypertext Construction.* 1995. pág. 87.
21. **Thüring, Hannemann y Haake.** *Hypermedia and Cognition: Designing for Comprehension.* 1995. págs. 57-66.
22. **Hammond.** *Learning with Hypertext: Problems, Principles and Prospects.* New York : s.n., 1993.

23. **Mira, José Joaquín, y otros, y otros.** *La Gestión por Procesos.* s.l. : Universidad Miguel Hernández de Helche.
24. **González Jorrín, Ing. Michel.** *Proceso de pruebas para la liberación de productos de software.* La Habana : Universidad de las Ciencias Informáticas, 2007.
25. **Núñez Sarmiento, Laura Isabel, Vélez Ramírez, Milena C. y Berdugo Correa, Carmen R.** *Aplicación de una metodología de mejora de procesos.* Barranquilla : Universidad del Norte. Colombia, 2004.
26. **Hermann, Jeffrey W y Lin, Edward.** *Supply Chain Simulation Modeling Using the Supply Chain Operations Reference Model.* 2003.
27. **Ortiz Forero, Juan Gabriel.** Gestipolis. [En línea] 28 de junio de 2010. [Citado el: 3 de mayo de 2011.] <http://www.gestipolis.com/administracion-estrategia-2/manufactura-esbelta.htm>.
28. **López, Gustavo.** *Metodología Six Sigma: Calidad Industrial.* Bogotá. Colombia : s.n.
29. **Grupo Kaizen.** Gestipolis. [En línea] septiembre de 2005. [Citado el: 3 de mayo de 2011.] <http://www.gestipolis.com/canales5/ger/gksa/39.htm>.
30. **Escalona Moreno, Ivan.** Gestipolis. [En línea] noviembre de 2003. [Citado el: 3 de mayo de 2001.] www.gestipolis.com/recursos/documentos/fulldocs/ger1/tociem.htm.
31. [En línea] 4 de abril de 2010. [Citado el: 5 de mayo de 2011.] sistemigramas.wordpress.com/2010/04/04/metodologiasistemassuaves.
32. **Centro de Excelencia para el Desarrollo de Proyectos Tecnológicos.** *Métricas de defectos.* La Habana : s.n., 2009.
33. **Torres, Mariela, Paz, Karim y Salazar, Federico G.** *Métodos de recolección de datos para una investigación.* 2009.

ANEXOS**Anexo 1 PLANILLA DE RESULTADOS DE LAS PRUEBAS EXPLORATORIAS****Nombre del Producto:****Versión del Producto:****Iteración:****Fecha:****Probador:****ACEPTADA: _____ RECHAZADA: _____****Indicadores a Evaluar:**

5. Se presenta la última versión realizada de los documentos.
6. Cantidad de documentos entregados
7. Se cumplen las reglas para la documentación pactadas en la capacitación inicial.
8. Los documentos entregados se corresponden con la última versión implementada.

Si el indicador número 1 no se cumple, no se pueden efectuar las revisiones.

Los restantes se evaluarán de la siguiente forma:

Si el por ciento de documentos que cumplen con los indicadores es menor que el 50% del total de los documentos, no se realizarán las revisiones.

Si el por ciento de documentos que cumplen con los indicadores se encuentra entre:

- 50%-60% el indicador tendrá peso 1.
- 60%-70% el indicador tendrá peso 2.
- 70%-80% el indicador tendrá peso 3.
- 80%-90% el indicador tendrá peso 4.

90%-100% el indicador tendrá peso 5.

| INDICADORES | Peso | | | | |
|---|------|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| Cantidad de documentos entregados | | | | | |
| Se cumplen las reglas para la documentación | | | | | |
| Correspondencia con la última versión implementada. | | | | | |

Una vez obtenidos los resultados de todos los indicadores, se procederá a calcular un resultado promedio (P) para obtener la evaluación final que determinará si se pueden comenzar o no las revisiones.

Este promedio se calculará de la siguiente manera:

- Se suman los pesos de cada indicadores (PI) y el valor obtenido se divide entre la cantidad de indicadores (CI):

$$P = \frac{\sum_{K=0}^n PI}{CI}$$

Mientras que el valor obtenido luego del cálculo del promedio, sea menor que 4, no se pueden comenzar las revisiones.

Anexo 2 CRITERIOS PARA DETENER LAS REVISIONES**Nombre del Producto:****Versión del Producto:****Iteración:****Fecha:****Probador:**

| No. | CRITERIOS | PRESENCIA |
|-----|---|-----------|
| 1 | Incoherencia significativa entre lo plasmado en el documento de requisitos y los documentos que se revisan. | |
| 2 | La cantidad de faltas de ortografía excede la cantidad de páginas del documento. | |
| 3 | A partir de la segunda iteración persisten al menos dos defectos significativos de iteraciones anteriores. | |
| 4 | Se observan los mismos tipos de defectos ya señalados en iteraciones anteriores en otros o mismos lugares donde fueron detectados inicialmente. | |

Para el criterio número 1 el probador debe informar los nombres de los documentos donde se encontró el defecto.

Para el criterio número 2 el probador debe brindar la siguiente información:

| Nombre del Documento | Cantidad de Faltas de Ortografía |
|----------------------|----------------------------------|
| | |
| | |

Para el criterio número 3 el probador debe brindar la siguiente información:

| Nombre del documento | Defecto | Iteración en que se identificó por primera vez | Localización del defecto |
|----------------------|---------|--|--------------------------|
| | | | |
| | | | |

Para el criterio número 4 el probador debe brindar la siguiente información:

| Nombre del documento | Defecto | Iteración en que se identificó por primera vez | Localización del defecto |
|----------------------|---------|--|--------------------------|
| | | | |
| | | | |

Anexo 3 PLANILLA DE NO CONFORMIDADES DE LAS REVISIONES

Nombre del Producto:

Versión del Producto:

Iteración:

Fecha:

Probador:

| Nombre del Documento | Descripción de la No Conformidad | Localización de la No Conformidad. | Significancia (Alta (A), Media (M), Baja (B)) |
|----------------------|----------------------------------|------------------------------------|---|
| | | | |

| | | | |
|--|--|--|--|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Anexo 4 PLANILLA DE NO CONFORMIDADES

Nombre del Producto:

Versión del Producto:

Nombre del Documento:

Cantidad de Páginas a revisar:

Iteración:

Fecha:

Probador:

| Tipo | Descripción de la No Conformidad | Localización de la No Conformidad. | Significancia (Alta (A), Media (M), Baja (B)) | Procede (Si o No) |
|------|----------------------------------|------------------------------------|---|-------------------|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

Anexo 5 LISTA DE CHEQUEO EMPLEADA PARA EVALUAR LA MUESTRA D

| Aspectos a Evaluar |
|--|
| ¿En la introducción se describen: los objetivos que persiguen, el problema que resuelve el producto y lo que pretende lograr al confeccionar el manual de usuario? |
| ¿Se entiende perfectamente las funcionalidades del sistema? |
| ¿Se utilizan imágenes para describir el manual? |
| ¿Las funcionalidades especificadas abarcan todos los requisitos del sistema? |
| ¿Se corresponden las funcionalidades especificadas con las recogidas en los requisitos del sistema? |
| ¿Ha identificado errores ortográficos? |
| ¿Se entiende claramente lo que se ha especificado en el documento? |
| ¿El número de página que aparece en el índice coincide con el contenido que se refleja realmente en dicha página? |
| ¿El total de páginas que aparecen en las reglas de confidencialidad coincide con el total de páginas que tiene el documento? |
| ¿Se encuentra actualizado el control de versiones del manual de usuario? |