

Universidad de las Ciencias Informáticas

“Facultad 3”



Título:

“Procedimiento para estimar tiempo y esfuerzo en las pruebas de liberación del laboratorio de calidad de la Universidad de las Ciencias Informáticas”.

*Trabajo de Diploma para optar por el título de
Ingeniero Informático.*

Autores: Tamara Castro Castro
Yamisley Rodríguez Faife

Tutores: Ing. Violena Hernández Aguilar
Ing. Yohandy Iglesias Cruz
Ing. Delvis Echeverría Pérez

**Ciudad de La Habana, mayo 2009.
“Año del 50 Aniversario del triunfo de la Revolución.”**

Declaración de Autoría

Declaración de Autoría:

Declaramos ser autoras del presente trabajo y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Tamara Castro Castro

Autores:

Yamisley Rodriguez Faife

Autores:

Ing. Violená Hernández Aguilar.

Tutora

Ing. Yohandy Iglesias Cruz

Tutor

Ing. Delvis Echeverría Pérez

Tutor

*La Imaginación es más
importante que el Conocimiento.*

Albert Einstein

Opinión de Tutores

Opinión de los Tutores:

Título: “Procedimiento para estimar duración y esfuerzo en las pruebas de liberación del laboratorio de calidad de la Universidad de las Ciencias Informáticas”.

Autores: Yamisley Rodríguez Faife.

Tamara Castro Castro.

Los tutores del presente Trabajo de Diploma consideran que durante su ejecución las estudiantes mostraron las cualidades que a continuación se detallan.

Tutora: Ing. Violena Hernández Aguilar.

Tutor: Yohandy Iglesias Cruz.

Tutor: Delvis Echeverría Pérez.

Opinión de Tutores

Por todo lo anteriormente expresado se considera que las estudiantes están aptas para ejercer como Ingeniero Informático; y se propone que se le otorgue al Trabajo de Diploma la calificación de ____.

Ing. Violena Hernández Aguilar.

Firma

Ing. Yohandy Iglesias Cruz.

Firma

Ing. Delvis Echeverría Pérez.

Firma

Fecha

Datos de Contacto

Datos de Contacto:

Tutora: Ing. Violena Hernández Aguilar. Correo electrónico: violena@uci.cu

- Ingeniera Informática del Instituto Superior Politécnico José Antonio Echeverría (2005).
- Profesora de la Universidad de las Ciencias Informáticas, en la Disciplina de Ingeniería y Gestión de Software desde el año 2005.
- Cuenta con 3 años de trabajo en la Educación Superior.
- Cursa la Maestría en Gestión de Proyectos Informáticos.
- Se desempeña laboralmente como Especialista General de la Dirección de Calidad de la Infraestructura Productiva de la UCI.

Tutor: Ing. Yohandy Iglesias Cruz. Correo electrónico: yiglesias@uci.cu

- Ingeniero Informático de la Universidad de las Ciencias Informáticas (2008).
- Profesor de la Universidad de las Ciencias Informáticas, en la Disciplina de Idioma Extranjero desde el año 2008.
- Cuenta con 1 año de trabajo en la Educación Superior.
- Cursa la Maestría en Gestión de Proyectos Informáticos.
- Se desempeña laboralmente como Especialista de Mercado en la Dirección de Gestión de Proyectos de ALBET.

Tutor: Ing. Delvis Echeverría Pérez. Correo electrónico: decheverria@uci.cu

- Ingeniero en Ciencias Informáticas de la Universidad de las Ciencias Informáticas (2007).
- Profesor de la Universidad de las Ciencias Informáticas, en la Disciplina Sistemas de Bases de Datos y Gestión de Software desde el año 2007.
- Cuenta con 2 años de trabajo en la Educación Superior.
- Cursa la Maestría en Calidad de Software.
- Se desempeña laboralmente como Especialista General de la Dirección de Calidad de la Infraestructura Productiva de la UCI.

Agradecimientos

A mis padres Leo y Ena, por la vida que me han regalado llena de amor y cariño. Por ser mis guías, ídolos y ser lo que más quiero en este mundo.

A mi hermana Taymi "tata" y a mi cuñado Héctor, por mortificarlos tanto y por darme seguridad y confianza.

A mis abuelos, porque sé que me cuidan y guían.

A mi superamiga y segunda hermana Elizabeth, porque junto a ella aprendí el significado de la palabra amistad.

A mi superhéroe Pepone, por joderlo cada vez que se me antoja y enseñarme lo que es el "nirvana".

A mi compañera de tesis "Fifi", por estar siempre a mi lado en los buenos y malos momentos.

A Violena, por su dedicación, apoyo y paciencia en este trayecto.

A Yohandy, por ser el "sácame de apuros" en todo momento.

A Delvis, por hacer más apacibles los momentos de tensión.

A toda mi familia, porque sé que siempre me han tenido presente y han vivido junto a mí cada momento importante en mi vida.

A todas mis amistades que compartieron trabajo, amor, sonrisas, consejos y momentos inolvidables.

Gracias a todos.

Tamara.

Agradecimientos

A mi abuela Milagros, por ser lo más bello del mundo, por todos sus desvelos, por el amor inmenso, a ella le debo todo lo que soy.

A mi papá, por sus enseñanzas, por darme la calma cuando la necesito y la fortaleza siempre.

A mi madrastra, por tratarme como una hija más, gracias por ser mi madre.

A mi mamá, por traerme al mundo, y posibilitar mi desempeño en la vida.

A mis hermanas, por todo su amor y apoyo, a mi hermano, por ser mi inspiración, mi razón de ser. A todos ellos por ser la alegría de mi vida.

A mi abuela Haydée, por su amor inmenso, sus consejos, y por quererme tanto como yo a ella.

A mi tía Omaidá, por el cariño durante toda mi vida, por su preocupación y dedicación, por ser mi amiga.

A mis tíos Roberto, Humberto, Luis y Yolexy, por considerarme una hija más, ayudarme y confiar en mí.

A mis primos, por su apoyo y entusiasmo, por reunirnos cada año y ser la alegría de mi abuela.

A Tamara, por aceptar ser mi compañera de tesis, por entenderme, y ser siempre mi refugio, gracias por ayudarme y ser mi guía.

Agradecimientos

A Violena, primero que nada por aceptar ser mi tutora, por la gran ayuda que me ha brindado en el desarrollo de este trabajo, por su dedicación, por darme serenidad en los momentos difíciles, por ser alguien a quien admiro, respeto y siempre le estaré agradecida.

A Yohandy, por el apoyo diario, por extenderme su mano y estar a mi lado en los momentos más difíciles, por escuchar mis lamentos, por estar siempre pendiente a mi, por comprenderme, soportarme y quererme, gracias por todos los momentos de felicidad, gracias por permitir que sea una parte pequeña de tu vida.

A Delvis, por poner la alegría en cada reunión de tesis, y por transmitirme su buen humor.

A Numidia, mi profesora del pre-universitario, por su amor y cariño, por acogerme como su familia.

A Nilda, por la dulzura de sus besos, por ser la persona más especial que he conocido, por ser mi mejor suegra, mi tercera mamá.

A Liuver, por todo su amor y comprensión, gracias por existir.

A la UCI y a las amistades que he encontrado en ella, a todos mis conocidos, a mi grupo de siempre, por los buenos y malos momentos que hemos pasado juntos.

Gracias a todos.

Yamisley.

Dedicatoria

A mis padres, Leo y Ena.

A mi hermana Taymi y mí cuñado Héctor.

A mi amiga Eliza.

Porque sin ellos no hubiese logrado llegar hasta aquí.

Tamara.

A mi abuela, por su esfuerzo y sacrificio.

A mis padres, hermanos y toda mi familia.

*A todas aquellas personas que me han ayudado a salir adelante,
y han hecho posible la realización de este sueño.*

Yamisley.

Resumen

Resumen

En la actualidad la industria del software ha evolucionado de forma extraordinaria, existe tendencia al crecimiento del volumen y complejidad de los productos, exigiéndose mayor calidad y productividad en menos tiempo. Es de vital importancia la realización de pruebas al producto para determinar su correctitud, completitud e integridad, estimar el tiempo y esfuerzo de estas pruebas es indispensable en el desarrollo de un software, ya que una estimación a la ligera puede traer consigo atrasos en la entrega y molestias al cliente que espera recibir un producto en tiempo y con la calidad adecuada.

La investigación surgió por la necesidad de contar con un procedimiento que permitiera estimar de forma eficiente el tiempo y esfuerzo de las pruebas de liberación del laboratorio de Calidad Central, pues se basan en la experiencia de los especialistas para hacer dicha estimación. El presente trabajo propone un procedimiento para estimar tiempo y esfuerzo de las pruebas de liberación en proyectos que utilicen metodología Rational Unified Process (RUP) en el laboratorio de Calidad Central, que permitirá a los especialistas seguir un patrón estándar para estimar dichas pruebas.

Palabras claves: estimación, software, pruebas de liberación, tiempo, esfuerzo, procedimiento, calidad.

Tabla de Contenido

Tabla de contenido:

TABLA DE CONTENIDO:	XII
INTRODUCCIÓN	1
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.....	6
1.1 Introducción.	6
1.1.1 El papel de las pruebas en el proceso de desarrollo del software.....	6
1.1.2 Objetivos de las Pruebas.	7
1.1.3 Principios de la prueba.....	7
1.1.4 Roles en pruebas.....	8
1.2 Importancia de las pruebas.	8
1.2.1 La importancia de la detección oportuna.	9
1.2.2 ¿Qué ventajas tiene la creación de pruebas para el desarrollo de software?.....	9
1.3 Estrategia de las pruebas.....	10
1.3.1 Objetivos de la estrategia de prueba.	11
1.3.2 Definir una estrategia de prueba correcta.	12
1.4 Niveles de prueba.	12
1. Prueba de desarrollador.	12
2. Prueba independiente.....	13
3. Prueba de unidad.	13
4. Prueba de integración.	14
5. Prueba del sistema.	15
7. Prueba de aceptación.	16
1.5 Tipos de prueba.	16
1.6 Métodos de prueba.	19
1.7 Estimación en prueba.	22
1.7.1 Técnicas de estimación del proceso de desarrollo de software.....	22
1.8 Pruebas en la UCI.....	26
1.9 Propuesta de solución.	27
1.10 Conclusiones Parciales	28
CAPÍTULO II: PROPUESTA DE SOLUCIÓN.....	29
2.1 Introducción.	29
2.2 Propuesta del procedimiento.....	29
2.3 Conclusiones Parciales	50
CAPÍTULO III: VALIDACIÓN DEL PROCEDIMIENTO.....	51
3.1 Introducción.	51
3.2 Métodos de expertos.....	51

Tabla de Contenido

3.3 Método Delphi.	52
3.4 Validación del procedimiento.	54
3.4.1 Proceso de selección del grupo de expertos.	54
3.4.2 Elaboración del Cuestionario.	57
3.4.3 Resultados de la validación.	58
3.5 Conclusiones parciales.	68
CONCLUSIONES	69
RECOMENDACIONES:	70
BIBLIOGRAFÍA	71
ANEXOS	74
GLOSARIO DE TÉRMINOS	97

Introducción

Introducción

La informática es la ciencia encargada del tratamiento automático de la información, disciplina que se aplica a numerosas y variadas áreas del conocimiento o la actividad humana, todas a través del producto software. Su evolución en crecimiento trae consigo un aumento de software en los mercados, creando la competitividad y es aquí donde aparece la calidad como factor imprescindible en la selección de un producto. El cliente cada vez exige más en menor tiempo y es donde se vive el dilema Calidad por Tiempo. (Clayton, 2005)

Durante la planeación del desarrollo de un software se plantea la importancia de la construcción del plan de aseguramiento de la calidad y se comienza a detallar el plan de pruebas técnicas, el cual cubrirá todos los productos que se consideren susceptibles de ser revisados o probados (RUP, 2003). Para lo cual es necesario utilizar los estándares, métodos, técnicas y herramientas para la validación y verificación de sistemas de software desde su fase inicial hasta la liberación y entrega.

Un producto software requiere de varios tipos de pruebas con diferentes métodos a distintos niveles, para probar que está listo para su liberación y entrega final (Pressman, 2005). Las pruebas de liberación son realizadas al software antes de su entrega al cliente para que este compruebe que el mismo cumple los requisitos pactados (pruebas de aceptación) y las lleva a cabo un grupo externo al equipo de desarrollo. (Calisoft, 2008) Una estimación a la ligera de esfuerzo y tiempo de duración de estas pruebas trae consigo el incumplimiento en la entrega del producto.

Cuba se empeña desde hace algunos años para que el sustento de su economía esté basado en las producciones intelectuales y en la actualidad el campo de la informática está dando grandes pasos de avance, rama que puede aportar mucho en los ingresos del país por conceptos de producción de software. Dentro de la Industria Cubana del Software juega un papel esencial la Universidad de las Ciencias Informáticas (UCI) para llegar a alcanzar este objetivo, pues como universidad de nuevo tipo, intenta convertirse en líder nacional de este campo.

La UCI desde sus inicios comenzó a insertarse con mucha fuerza y de forma creciente en la industria del software y en solo siete años ha ganado en seriedad, organización y ha alcanzado prestigio tanto a nivel nacional como internacional, muestra de ello son los disímiles proyectos en los que se está trabajando, tanto de exportación como para las diferentes empresas, entidades o ministerios del país.

Introducción

Muchos de los productos de software elaborados por la universidad no tenían la calidad requerida, ante tal situación se crea la Dirección de Calidad Central, surgiendo así las pruebas de liberación como un nuevo nivel de pruebas. Las pruebas de liberación son un concepto propio del centro y se llevan a cabo antes de realizar las pruebas de aceptación. Los responsables de la realización de las mismas son los especialistas del laboratorio de pruebas de la Dirección de Calidad de la universidad, los cuales realizan pruebas funcionales y no funcionales y a partir de criterios de criticidad definidos liberan las aplicaciones o abortan las pruebas debido a la mala calidad del software revisado. (Calisoft, 2008)

Actualmente en la realización de las pruebas de liberación no se usa ningún método o procedimiento formal que permita calcular la estimación correcta de esfuerzo y tiempo de duración de las mismas, lo cual trae consigo incertidumbres con respecto al tiempo de duración y al esfuerzo a emplear para llevar a cabo las pruebas de aceptación. Para la estimación de tiempo y esfuerzo, los especialistas se basan en su experiencia para la realización de las pruebas, por lo que no hay una forma estándar de estimar.

Ver Anexo 1.

Luego el **problema** que se plantea resolver esta investigación es:

¿Cómo estimar de forma eficiente el tiempo y esfuerzo empleado en las pruebas de liberación en el laboratorio de Calidad Central de la universidad?

Por tanto el **objeto de estudio** es el proceso de pruebas de software.

Teniendo en cuenta lo anteriormente planteado el **objetivo general** de esta investigación es:

Proponer un procedimiento que establezca de forma eficiente la estimación de tiempo y esfuerzo de las pruebas de liberación (en horas x hombres), en el laboratorio de Calidad Central de la universidad.

Las tareas que se proponen para dar cumplimiento al objetivo son:

- Estudiar el proceso de prueba en el mundo y compararlo con el existente en el laboratorio de pruebas de liberación de software.
- Proponer un procedimiento para la estimación de tiempo y esfuerzo en las pruebas de liberación en la UCI.
- Validar el procedimiento usando criterio de expertos.

Introducción

Por tanto el **campo de acción** es la estimación de tiempo y esfuerzo en las pruebas de liberación de los proyectos de la Universidad de las Ciencias Informáticas.

Se plantean como guía de la investigación las siguientes **preguntas científicas**:

- ¿La estimación de las pruebas que se realizan en el laboratorio de Calidad Central se acercan a la realidad?
- ¿Se tienen en cuenta la experiencia del personal que lleva a cabo la estimación del esfuerzo del diseño e implementación de las pruebas?
- ¿Se debería tener en cuenta la complejidad de los artefactos de entrada a las pruebas para realizar una estimación más precisa?
- ¿Al realizar la estimación de esfuerzo es necesario tener en cuenta los riesgos técnicos y de recursos?
- ¿Si se estandariza la estimación será posible obtener una estimación que se aproxime al esfuerzo real?

Métodos usados en la investigación.

Métodos teóricos:

- Analítico – sintético. Se analizaron las teorías, datos y documentos, para extraer los elementos más importantes que se relacionan con el objeto de estudio.
 - Análisis: permite la división mental del fenómeno en sus múltiples relaciones y componentes para facilitar su estudio.
 - Síntesis: establece mentalmente la unión entre las partes previamente analizadas, posibilita descubrir sus características generales y las relaciones esenciales entre ellas.
- Histórico – lógico. Ayudó a comprobar teóricamente datos históricos que han ido evolucionando en un período de tiempo. Permite estudiar de forma analítica la trayectoria histórica real de los fenómenos, su evolución y desarrollo.

Ambos métodos fueron utilizados para la recuperación de información la cual sirvió para la determinación de rangos, valores y elementos a tener en cuenta en el procedimiento. El análisis de estos datos por parte de los especialistas del centro ayudó al perfeccionamiento del procedimiento.

Introducción

Métodos empíricos:

- Entrevista.
- Encuesta.

Con el propósito de obtener información, experiencias, ideas y puntos de vistas que contribuyan al desarrollo de la investigación y aporten conocimientos específicos del tema se utilizaron los métodos empíricos. Con el fin de obtener información sobre el proceso de pruebas en la UCI y para determinar la situación problemática en la cual está enmarcado el proceso de pruebas de liberación se puso en práctica la entrevista. Además la encuesta se utilizó para la validación del procedimiento y de los elementos que se tuvieron en cuenta a la hora de su realización.

Aportes teóricos de la investigación.

El procedimiento que se propone permite mejorar la estimación de tiempo y esfuerzo en el proceso de pruebas de liberación en el laboratorio de Calidad Central, permitiendo así un mejor trabajo, cumplimiento y entrega del producto software.

Significación práctica de la investigación.

La implantación de este procedimiento en el laboratorio de Calidad Central permitirá que las pruebas de liberación sean estimadas de acuerdo con las características del producto y se pueda liberar con el máximo de calidad requerida.

La tesis está estructurada de la siguiente forma:

- **Introducción.**
- **Tres Capítulos.**
- **Conclusiones.**
- **Recomendaciones.**

En el **Capítulo I** se realiza un análisis de los conceptos de calidad, prueba y estimación, dando una panorámica de todo lo referente a niveles, tipos, métodos y estrategias de pruebas, así como la situación de este flujo de trabajo en la universidad. Se explican algunas de las tendencias actuales para estimar

Introducción

tiempo y esfuerzo en las pruebas, concluyendo con los elementos fundamentales de la propuesta de solución.

En el **Capítulo II** se establecen los pasos para llevar a cabo el procedimiento propuesto, así como las fórmulas para el cálculo del esfuerzo y el tiempo de diseño e implementación de las pruebas.

En el **Capítulo III** se valida la solución propuesta usando el método de expertos, específicamente método delphi para lo cual se escogieron 8 expertos y se realizaron varias rondas de preguntas hasta llegar a un consenso en cuanto a la solución.

Capítulo 1: Fundamentación Teórica

Capítulo I: Fundamentación teórica.

1.1 Introducción.

Para la elaboración de un producto software es necesaria la utilización de metodologías, las cuales guían el proceso de desarrollo. La universidad utiliza el Proceso Unificado de Rational (RUP), junto con el Lenguaje Unificado de Modelado (UML), que constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP esta conformado por 6 flujos ingenieriles: Modelamiento del negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba y Despliegue.

Las pruebas a pesar de tener un apartado dentro de la metodología RUP, requiere de su aplicación durante toda la vida del software. Esto posibilita erradicar errores desde sus comienzos, que más tarde traerían innumerables gastos y baja calidad en la entrega al cliente. Un software de alta calidad es esencial para el éxito de los sistemas críticos para la seguridad, (como el control del tráfico aéreo, la orientación de misiles, o los sistemas médicos), donde una anomalía puede ocasionar grandes pérdidas humanas y materiales.

1.1.1 El papel de las pruebas en el proceso de desarrollo del software.

El Flujo de Trabajo de Pruebas aparece a lo largo de todo el proceso de desarrollo del software y se hace a través de validaciones y verificaciones, las validaciones se llevan a cabo con el cliente, con el fin de ver si la construcción es lo que realmente el cliente desea, y las verificaciones se hacen con el objetivo de verificar si el artefacto está bien construido, pueden existir dos tipos de verificaciones: las estáticas que se llevan a cabo usando listas de chequeo, revisiones técnicas formales y auditorías, y las verificaciones dinámicas, que es la realización de la prueba, las cuales son los diferentes procesos que se deben realizar durante un desarrollo, con el objetivo de asegurar completitud, correctitud y calidad en un producto software. (RUP, 2003)

Pruebas.

Actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente. (Entorno Virtual de Aprendizaje, 2006) La prueba como disciplina actúa para

Capítulo 1: Fundamentación Teórica

proveer servicios a otras disciplinas en muchos aspectos. Las pruebas se centran principalmente en la evaluación o la valoración de la calidad del producto, hecho que se lleva a cabo mediante las prácticas: (RUP, 2003)

- Buscar y documentar los defectos en la calidad del software.
- Opinar sobre la calidad percibida del software.
- Validar y demostrar las suposiciones efectuadas en las especificaciones de diseño y requisitos con una demostración concreta.
- Validar que el producto software funciona según lo diseñado.
- Validar que los requisitos se han implementado de forma adecuada.

1.1.2 Objetivos de las Pruebas.

Glen Myers (Myers, 1979) establece varias normas que pueden servir adecuadamente como objetivos de la prueba:

- La prueba es un proceso de ejecución de un programa con la intención de descubrir un error.
- Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
- Una prueba tiene éxito si descubre un error no detectado hasta entonces.

1.1.3 Principios de la prueba.

Los principios más importantes planteados por Davis (Davis, 1995), Edward (Edward, 1986) y Myers (Myers, 1979) son:

- La prueba puede ser usada para mostrar la presencia de errores, pero nunca de su ausencia.
- La principal dificultad del proceso de prueba es decidir cuando parar.
- Evitar casos de prueba no planificados, no reusables y triviales a menos que el programa sea verdaderamente sencillo.
- Una parte necesaria de un caso de prueba es la definición del resultado esperado.
- Los casos de prueba tienen que ser escritos, no solo para condiciones de entradas válidas y esperadas, sino también para condiciones no válidas e inesperadas.
- Los casos de prueba tienen que ser escritos para generar las condiciones de salida deseadas.

Capítulo 1: Fundamentación Teórica

- El número de errores sin descubrir es directamente proporcional al número de errores descubiertos.
- Las pruebas deberían empezar por “lo pequeño” y progresar hacia “lo grande”.
- Con la excepción de las pruebas de unidad e integración, un programa deberá ser probado por la persona u organización que lo desarrolló.
- Asigna el programador más creativo a la prueba.

1.1.4 Roles en pruebas.

Definición del comportamiento y la responsabilidad de un individuo o grupo de individuos que trabajan juntos como un equipo, en el contexto de una empresa. (RUP, 2003)

- Administrador de Prueba: Es el responsable del éxito de la prueba, este rol involucra defensor de prueba y calidad, planificación y administración de recursos, resolución de problemas que impiden las pruebas.
- Analista de Prueba: Es el responsable de identificar y definir las pruebas requeridas, monitorear el progreso de la prueba, el resultado en cada ciclo de prueba y evaluar la calidad total experimentada como un resultado de las actividades de prueba. Este rol lleva la responsabilidad para representar apropiadamente las necesidades de los stakeholder que no tienen representación regular y directa en el proyecto.
- Diseñador de Prueba: Es el responsable de definir el método de prueba y asegurar su implementación exitosa. El rol incluye identificar técnicas apropiadas, herramientas e instrucciones para implementar las pruebas necesarias y encauzar los recursos correspondientes para las pruebas.
- Probador: Es el responsable durante las actividades principales de las pruebas, el cual incluye la conducción de las pruebas necesarias y el registro del resultado de la prueba.

1.2 Importancia de las pruebas.

Las pruebas de software están poco consideradas a pesar de su importancia estratégica. La robustez del software es un elemento fundamental, pero que pocas personas aplican en sus proyectos. El proyecto promedio tiene semanas destinadas a la prueba, principalmente en las semanas antes del despliegue. Por supuesto, la mayoría del software termina retrasado y por encima del presupuesto, y la prueba es la

Capítulo 1: Fundamentación Teórica

primera línea que se reduce o se elimina. La robustez no es un módulo que puede ser incrustado a un lado de un sistema preexistente, es mucho más efectivo en costos si se desarrolla un software robusto, al cual se le realicen pruebas que aboguen por la calidad desde el día uno. (John V, 2008)

1.2.1 La importancia de la detección oportuna.

En la cadena de valor del desarrollo de un software específico, el proceso de prueba es clave a la hora de detectar errores o fallas. Conceptos como estabilidad, escalabilidad, eficiencia y seguridad se relacionan a la calidad de un producto bien desarrollado. Las aplicaciones de software han crecido en complejidad y tamaño, y por consiguiente también en costos. Hoy en día es crucial verificar y evaluar la calidad de lo construido, para minimizar el costo de su reparación. Mientras antes se detecte una falla, más barata es su corrección. (Caldas, 2006)

Las pruebas no se deben dejar para el final de la fase de construcción del software, se deben empezar a realizar desde el mismo flujo de requerimientos, ya que desde un principio puede incurrirse en malas interpretaciones de las "reglas del negocio", lo que finalmente tendrá como consecuencia, incongruencia entre lo que el cliente quiere y lo que se ha desarrollado.

1.2.2 ¿Qué ventajas tiene la creación de pruebas para el desarrollo de software?

La importancia de las pruebas en cualquier proceso de desarrollo es señalado claramente tanto en procesos clásicos, modelo en cascada, y otros más actuales, como el proceso unificado de desarrollo o metodologías ágiles. El planteamiento actual es que las pruebas son parte integral e incluso dirigen el desarrollo. (Caldas, 2006)

Las principales ventajas que trae consigo la realización de pruebas en un desarrollo de software son:

- Reducen la posibilidad de agregar defectos al software. Si hay que realizar una adición de características requeridas por el cliente y se ve que ya no funcionan bien algunas de las cosas que anteriormente servían, se puede inferir la nueva funcionalidad, que es la que contiene defectos, por lo que no hay necesidad de realizar modificaciones a los componentes realizados anteriormente.
- Reducen la posibilidad de encontrar defectos en funcionalidades ya implementadas.

Capítulo 1: Fundamentación Teórica

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

1.3 Estrategia de las pruebas.

Una estrategia de prueba del software integra las técnicas de diseño de casos de prueba en una serie de pasos bien planificados que llevan a la construcción correcta del software. La estrategia proporciona un mapa que describe los pasos que hay que llevar a cabo como parte de la prueba, cuando se deben planificar y realizar esos pasos, y cuanto esfuerzo, tiempo y recurso se va a requerir, por tanto, cualquier estrategia de prueba debe incorporar la planificación de la prueba, el diseño de casos de pruebas, la ejecución de la pruebas y la agrupación y evaluación de los datos resultantes. (Pressman, 2005)

Una estrategia de pruebas de software debe ser lo suficientemente flexible para promover la creatividad y adaptabilidad necesaria para adecuar las pruebas a todos los grandes sistemas basados en software. Al

Capítulo 1: Fundamentación Teórica

mismo tiempo, la estrategia debe ser lo suficientemente rígida para promover un seguimiento razonable de la planificación y la gestión a medida que progresa el proyecto. (Pressman, 2005)

Las características generales que debe seguir una estrategia de prueba son:

- La prueba comienza en el nivel de módulo y trabaja “hacia afuera”.
- En diferentes puntos son adecuadas a la vez distintas técnicas de prueba.
- La prueba la realiza la persona que desarrolla el software y (para grandes proyectos) un grupo de pruebas independiente.
- La prueba y la depuración son actividades diferentes.
- Una estrategia de prueba para el software debe constar de pruebas de bajo nivel, que verifiquen que todos los pequeños segmentos de código fuente sean implementados correctamente, así como de pruebas de alto nivel que validen las principales funciones del sistema frente a los requisitos del cliente. (Pressman, 2005)
- Una estrategia debe proporcionar una guía al profesional y un conjunto de hitos para el jefe de proyecto. Debido a que los pasos de la estrategia de prueba se dan a la vez cuando aumenta la presión de los plazos fijados, se debe poder medir el progreso y los problemas deben aparecer lo antes posible. (Pressman, 2005)

1.3.1 Objetivos de la estrategia de prueba.

Las estrategias de pruebas tienen los siguientes objetivos: (Pressman, 2005)

- Planificar las pruebas necesarias en cada iteración, incluyendo las pruebas de unidad, integración y las pruebas de sistema. Las pruebas de unidad y de integración son necesarias dentro de la iteración, mientras que las pruebas de sistema son necesarias sólo al final de la iteración.
- Diseñar e implementar las pruebas creando los casos de prueba que especifican qué probar, y cómo realizar las pruebas, creando, si es posible, componentes de prueba ejecutables para automatizar las pruebas.
- Realizar diferentes pruebas y manejar los resultados de cada prueba sistemáticamente. Los productos de desarrollo de software en los que se detectan defectos son probados de nuevo y posiblemente devueltos a otra etapa, como diseño o implementación, de forma que los defectos puedan ser arreglados.

Capítulo 1: Fundamentación Teórica

1.3.2 Definir una estrategia de prueba correcta.

Para definir una estrategia de prueba correcta es necesario: (Pressman, 2005)

- Especificar los requisitos del producto de manera cuantificable mucho antes de que comiencen las pruebas.
- Establecer los objetivos de la prueba de manera explícita.
- Comprender qué usuarios van a manejar el software y desarrollar un perfil para cada categoría de usuario.
- Desarrollar un plan de prueba que haga hincapié en la prueba de ciclo rápido.
- Construir un software robusto, diseñado para probarse a si mismo.
- Usar revisiones técnicas formales efectivas, como filtro antes de la prueba.
- Llevar a cabo revisiones técnicas formales para evaluar la estrategia de prueba y los propios casos de prueba.
- Desarrollar un enfoque de mejora continua al proceso de prueba.

1.4 Niveles de prueba.

Las pruebas se aplican a diferentes tipos de destinos, en fases o niveles diferentes de esfuerzo de trabajo. Estos niveles suelen distinguirlos los roles que están más capacitados para diseñar y dirigir las pruebas, donde las técnicas son más adecuadas para realizar la prueba en cada nivel. Es importante asegurarse que hay un equilibrio entre los diferentes esfuerzos de trabajo. (RUP, 2003)

1. Prueba de desarrollador.

La prueba de desarrollador indica los aspectos de diseño e implementación de la prueba más adecuada que debe llevar a cabo el equipo de desarrolladores, a diferencia de la prueba independiente. En la mayoría de los casos, la ejecución de la prueba se produce inicialmente con el grupo de pruebas de desarrollador que la diseñó e implementó; aunque es recomendable que los desarrolladores creen las pruebas de forma que estén disponibles para que las ejecuten grupos de pruebas independientes. Es recomendable que la prueba de desarrollador no cubra únicamente unidades independientes de pruebas aisladas.

Capítulo 1: Fundamentación Teórica

2. Prueba independiente.

Las pruebas independientes indican el diseño y la implementación de la prueba realizada más adecuadamente por alguien ajeno al equipo de desarrolladores. Puede considerarse esta distinción un superconjunto, que incluye validación y verificación independiente. En la mayoría de los casos, la ejecución de la prueba se produce inicialmente con el grupo de pruebas independientes que la diseñó e implementó; aunque los verificadores independientes deberían crear sus pruebas de forma que estén disponibles para que las ejecuten los grupos de pruebas de desarrollador. Las pruebas independientes se pueden ver de forma alternativa, como representaciones de pruebas basadas en las necesidades y las preocupaciones de varios interesados; por eso también se conocen como prueba de los interesados.

3. Prueba de unidad.

La prueba de unidad se centra en la verificación de los elementos más pequeños del software que se puedan probar. Normalmente, las pruebas de unidad se aplican a componentes representados en el modelo de implementación, para verificar que se cubren los flujos de control, los flujos de datos y que funcionan como se esperaba. El implementador realiza la prueba de unidad que se describen en la disciplina de implementación. Las pruebas que se dan como parte de la prueba de unidad, prueban la interfaz del módulo para asegurar que la información fluye de forma adecuada hacia y desde la unidad de programa que está siendo probada.

Se examinan las estructuras de datos locales para asegurar que los datos que se mantienen temporalmente conservan su integridad durante todos los pasos de ejecución del algoritmo. Se prueban las condiciones límites para asegurar que el módulo funciona correctamente en los límites establecidos como restricciones de procesamiento. Se ejercitan todos los caminos independientes (caminos básicos) de la estructura de control, con el fin de asegurar que todas las sentencias del módulo se ejecutan por lo menos una vez. Y, finalmente, se prueban todos los caminos de manejo de errores.

Durante la prueba de unidad, la comprobación selectiva de los caminos de ejecución es una tarea esencial. Se deben diseñar casos de prueba para detectar errores debidos a cálculos incorrectos, comparaciones incorrectas o flujos de control inapropiados. Las pruebas del camino básico y de bucles son técnicas muy efectivas para descubrir una gran cantidad de errores en los caminos.

Capítulo 1: Fundamentación Teórica

4. Prueba de integración.

Las pruebas de integración se realizan para garantizar que los componentes del modelo de implementación funcionan correctamente cuando se combinan para ejecutar un guión de uso. El destino de la prueba es un paquete o un conjunto de paquetes del modelo de implementación. A menudo, los paquetes que se combinan proceden de diferentes empresas de desarrollo. Las pruebas de integración exponen el estado incompleto o los errores de las especificaciones de la interfaz del paquete.

En algunos casos, los desarrolladores presuponen que otros grupos, como los verificadores independientes, son los encargados de realizar las pruebas de integración. Esta situación supone un riesgo para el producto de software y, en última instancia, para la calidad del software, puesto que:

- Las áreas de integración son un punto común de fallo del software.
- Las pruebas de integración realizadas por verificadores independientes suelen utilizar técnicas de caja negra y tratar componentes de software más grandes.

Un enfoque mejor, es considerar las pruebas de integración como responsabilidad tanto del desarrollador como de los verificadores independientes, e impedir que la estrategia de esfuerzo de prueba de los equipos no se superponga de forma significativa. La naturaleza exacta de esta superposición se basa en las necesidades del proyecto individual. Es recomendable que fomente un entorno donde los desarrolladores y los verificadores independientes del sistema compartan una visión única de la calidad.

Integración descendente.

La prueba de integración descendente es un planteamiento incremental en la construcción de la estructura de programas. Se integran los módulos moviéndose hacia abajo por la jerarquía de control, comenzando por el módulo de control principal (programa principal). Los módulos subordinados (subordinados de cualquier modo) a un módulo de control principal se van incorporando en la estructura, bien de forma primero-en-profundidad, o bien de forma primero-en-anchura.

Integración ascendente.

La prueba de la integración ascendente, como su nombre indica, empieza la prueba con los módulos atómicos (es decir, módulos de los niveles más bajos de la estructura del programa). Dado que los

Capítulo 1: Fundamentación Teórica

módulos se integran de abajo hacia arriba, el proceso requerido de los módulos subordinados siempre está disponible y se elimina la necesidad de resguardos.

Prueba de regresión.

Cada vez que se añade un nuevo módulo como parte de una prueba de integración, el software cambia. Se establecen nuevos caminos de flujo de datos, pueden ocurrir nuevas E/S y se invoca una nueva lógica de control. Estos cambios pueden causar problemas con funciones que antes trabajaban perfectamente. En el contexto de una estrategia de prueba de integración, la prueba de regresión es volver a ejecutar un subconjunto de pruebas que se han llevado a cabo anteriormente, para asegurarse de que los cambios no han propagado efectos colaterales no deseados.

Prueba de humo.

La prueba de humo es un método de prueba de integración que es comúnmente utilizado cuando se ha desarrollado un producto software empaquetado. Es diseñado como un mecanismo para proyectos críticos por tiempo, permitiendo que el equipo de software valore su proyecto sobre una base salida. En esencia, la prueba de humo comprende las siguientes actividades:

- Los componentes software que han sido traducidos a código, se integran en una construcción. Una construcción incluye ficheros de datos, librerías, módulos reutilizables, y componentes de ingeniería que se requieren para implementar una o más funciones del producto.
- Se diseña una serie de pruebas para descubrir errores que impiden a la construcción realizar su función adecuadamente. El objetivo será descubrir errores bloqueantes que tengan la mayor probabilidad de impedir al proyecto de software el cumplimiento de su planificación.
- Es habitual en la prueba de humo que la construcción se integre con otras construcciones y que se aplique una prueba de humo al producto completo. La integración puede hacerse bien de forma descendente (Top-down) o ascendente (bottom-up).

5. Prueba del sistema.

Normalmente, la prueba del sistema se realiza cuando el software funciona en su totalidad. Un ciclo vital repetitivo permite que las pruebas del sistema se realicen mucho antes, en cuanto se hayan implementado subconjuntos bien formados del comportamiento de guiones de uso. Normalmente, el destino son los elementos en funcionamiento de extremo a extremo del sistema.

Capítulo 1: Fundamentación Teórica

Estas pruebas caen fuera del ámbito del proceso de ingeniería del software y no las realiza únicamente el desarrollador del software. Sin embargo, los pasos dados durante el diseño del software y durante la prueba pueden mejorar enormemente la probabilidad de éxito en la integración del software en el sistema. La prueba del sistema, realmente, está constituida por una serie de pruebas diferentes cuyo propósito primordial es ejercitar profundamente el sistema basado en computadora.

Prueba de recuperación.

La prueba de recuperación es una prueba del sistema que fuerza el fallo del software de muchas formas y verifica que la recuperación se lleva a cabo apropiadamente. Si la recuperación es automática (llevada a cabo por el propio sistema) hay que evaluar la corrección de la inicialización, de los mecanismos de recuperación del estado del sistema, de la recuperación de datos y del proceso de re arranque. Si la recuperación requiere la intervención humana, hay que evaluar los tiempos medios de reparación (TMR) para determinar si están dentro de unos límites aceptables.

6. Pruebas de liberación.

Es una prueba que se hace antes de la aceptación, es una prueba de sistema, es decir, tiene lugar antes de que llegue al cliente, es realizada por personas externas a la organización, no puede ser el cliente porque no sabe cómo funcionan los procesos, o sea, el negocio. Se buscan errores desde afuera, es una prueba que se tomó en consideración en el laboratorio de Calidad Central.

7. Prueba de aceptación.

La prueba de aceptación del usuario es la última acción de prueba antes de desplegar el software. El objetivo de la prueba de aceptación es comprobar si el software está preparado y lo pueden utilizar los usuarios para realizar las funciones y tareas para las que se diseñó.

1.5 Tipos de prueba.

Las pruebas de software informático implican mucho más que la simple evaluación de las funciones, la interfaz y las características de tiempo de respuesta de un destino de la prueba, para lo cual deben implementarse y ejecutarse muchos tipos diferentes de pruebas. Cada tipo de prueba tiene un objetivo específico y una técnica de soporte, que esta a su vez se centra en la prueba de uno o varios atributos o características del destino de la prueba. (RUP, 2003)

Capítulo 1: Fundamentación Teórica

A continuación se listan los tipos de prueba basados en las dimensiones de calidad:

Funcionalidad

- **Prueba de función:** pruebas que se centran en la validación de funciones del destino de la prueba, proporcionan los guiones de uso, los métodos y los servicios necesarios. Esta prueba se implementa y se ejecuta en diferentes destinos de la prueba, incluidas las unidades, las unidades integradas, las aplicaciones y los sistemas.
- **Prueba de seguridad:** pruebas que se centran en garantizar que los datos del destino de la prueba (o sistemas) sólo son accesibles para los actores a los que se dirigen. Esta prueba se implementa y ejecuta en varios destinos de la prueba.
- **Prueba de volumen:** pruebas que se centran en la verificación de la capacidad del destino de la prueba para manejar grandes cantidades de datos, ya sean de entrada y salida o residentes, en la base de datos. La prueba de volumen incluye estrategias de prueba como la creación de consultas que devolverán el contenido completo de la base de datos, o que tendrán tantas restricciones que no devolverán ningún dato, o en las que la entrada de datos tiene la cantidad máxima de datos para cada campo.

Utilización

- **Prueba de utilización:** pruebas que se basan en:
 - factores humanos
 - estética
 - coherencia de la interfaz de usuario
 - ayuda en línea y según contexto
 - asistentes y agentes
 - documentación de usuario
 - materiales de formación
- **Prueba de integridad:** pruebas que se centran en la evaluación de la fuerza del destino de la prueba (resistencia a los errores) y la conformidad técnica del lenguaje, la sintaxis y la utilización de recursos. Esta prueba se implementa y se ejecuta en diferentes destinos de la prueba, incluidas las unidades y las unidades integradas.

Capítulo 1: Fundamentación Teórica

Fiabilidad

- **Prueba de estructura:** pruebas que se centran en la evaluación de la adherencia del destino de la prueba a su diseño y formación. Normalmente, esta prueba se realiza en aplicaciones habilitadas para web y garantiza que todos los enlaces están conectados, se muestra el contenido adecuado y no hay ningún contenido huérfano.
- **Prueba de tensión:** se trata de un tipo de prueba de fiabilidad que se centra en la evaluación de cómo responde el sistema en circunstancias anormales. Las tensiones del sistema pueden ser cargas de trabajo extremas, memoria insuficiente, servicios y hardware no disponible o recursos compartidos limitados. Estas pruebas suelen realizarse para saber mejor cómo y en qué áreas fallará el sistema, de forma que se puedan planificar y presupuestar los planes de contingencia y el mantenimiento de las actualizaciones con bastante antelación.
- **Prueba de puntos de referencia:** se trata de un tipo de prueba de rendimiento que compara el rendimiento de un destino de la prueba nuevo o desconocido con una referencia conocida, carga de trabajo y sistema.
- **Prueba de contienda:** pruebas que se centran en la validación de la capacidad del destino de la prueba para manejar de forma aceptable varias demandas del actor en el mismo recurso (registros de datos, memoria, entre otros).

Rendimiento

- **Prueba de carga:** se trata de un tipo de prueba de rendimiento que se utiliza para validar y evaluar la aceptabilidad de los límites operativos de un sistema bajo cargas de trabajo variables, mientras el sistema que se está probando permanece igual. En algunas variantes, la carga de trabajo permanece igual y se modifica la configuración del sistema que se está probando. Las medidas suelen tomarse en función del rendimiento de la carga de trabajo y el tiempo de respuesta de las transacciones en línea. Las variaciones de la carga de trabajo suelen incluir la emulación del pico y el promedio de cargas de trabajo que se producen dentro de la tolerancia operativa normal.
- **Perfil de rendimiento:** se trata de una prueba en la que se controla el perfil de tiempo del destino de la prueba, incluidos el flujo de la ejecución, el acceso de datos, las llamadas del sistema y de funciones para identificar y tratar los cuellos de botella de rendimiento y los procesos ineficaces.

Capítulo 1: Fundamentación Teórica

- **Prueba de configuración:** pruebas que se centran en garantizar que las funciones del destino de la prueba son las adecuadas en diferentes configuraciones de hardware y software. Esta prueba también se puede implementar como una prueba de rendimiento del sistema.

Capacidad de soporte

- **Prueba de instalación:** pruebas que se centran en garantizar que el destino de la prueba se instala correctamente en diferentes configuraciones de hardware y software, y en condiciones diferentes como por ejemplo, (espacio de disco insuficiente o interrupciones de la alimentación). Esta prueba se implementa y ejecuta en aplicaciones y sistemas.

1.6 Métodos de prueba.

Método Híbrido:

El Método Híbrido es el método de prueba más completo que se le puede aplicar a un software, ya que consiste en aplicar los siguientes métodos: (Lander JA, 1999)

- Método de Caja Negra (Análisis de Fronteras).
- Método de Caja Blanca (Cobertura Ciclomática de Caminos).
- Pruebas de Malicia.

➤ Prueba de Caja Negra.

El método de la caja negra se centra en los requisitos fundamentales del software y permite obtener entradas que prueben todos los requisitos funcionales del programa. Con este equipo de pruebas se intenta encontrar: (Pressman, 2005)

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

Técnicas de pruebas basadas en pruebas de caja negra.

- ✓ **Partición equivalente:** Pressman presenta la partición equivalente como un método de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se

Capítulo 1: Fundamentación Teórica

pueden derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar.

El objetivo de partición equivalente, es reducir el posible conjunto de casos de prueba en uno más pequeño, un conjunto manejable que evalúe bien el software. Se toma un riesgo porque se escoge no probar todo. Así que se necesita tener mucho cuidado al escoger las clases.

- ✓ **Análisis de valores límite:** Los errores tienden a darse más en los límites del campo de entrada que en el centro. Por ello, se ha desarrollado el análisis de valores límites (AVL) como técnica de prueba. El análisis de valores límite lleva a una elección de casos de prueba que ejerciten los valores límite.

El análisis de valores límite es una técnica de diseño de casos de prueba que completa a la partición equivalente. En lugar de seleccionar cualquier elemento de una clase de equivalencia, el AVL lleva a la elección de casos de prueba en los extremos de la clase. En lugar de centrarse solamente en las condiciones de entrada, el AVL obtiene casos de prueba también para el campo de salida.

- ✓ **Gráfica Causa-efecto.** La gráfica Causa-efecto representa una ayuda gráfica en seleccionar, de una manera sistemática, un gran conjunto de casos de prueba. Tiene un efecto secundario beneficioso en precisar estados incompletos y ambigüedades en la especificación.

➤ **Prueba de Caja Blanca.**

El método de caja blanca se centra en los requisitos no funcionales, basándose en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de pruebas que ejerciten conjuntos específicos de condiciones y/o bucles. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado. (Pressman, 2005)

Capítulo 1: Fundamentación Teórica

Técnicas de pruebas basadas en pruebas de caja blanca.

- ✓ **Camino básico:** Es una técnica de prueba de caja blanca, permite obtener una medida de la complejidad lógica de su diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución, garantizando que se ejecute por lo menos una vez cada sentencia del programa.
- ✓ **Prueba de condición:** Es un método de diseño de casos de prueba que ejercita las condiciones lógicas contenidas en el módulo de un programa, centrándose en las pruebas de cada una de las condiciones del programa, cuyo propósito es detectar no solo los errores en las condiciones de un programa sino también otros errores en dichos programas.
- ✓ **Prueba de flujo de datos:** Selecciona caminos de pruebas de un programa de acuerdo con la ubicación de las definiciones, y los usos de las variables del programa, son útiles para seleccionar caminos de prueba de un programa que contenga sentencias if o de bucles anidados.
- ✓ **Pruebas de bucles:** Se centra exclusivamente en la validez de las construcciones de bucles. Se pueden definir 4 clases diferentes de bucles: bucles simples, bucles concatenados, bucles anidados y bucles no estructurados.

➤ Pruebas de Malicia.

Las Pruebas de Malicia consisten en ejecutar acciones que no están contempladas en el diseño dinámico del software en cuestión, tratando de identificar los fallos del sistema con eventos que puede producir el usuario pero que no siguen los lineamientos del programa. En este método se obvia intencionalmente la estructura del programa, a diferencia de la Cobertura Ciclomática de Caminos, ya que el ingeniero de pruebas debe colocarse en la posición del usuario. Igualmente que el método de Caja Negra, es recomendable que éste procedimiento lo realice una persona diferente al que realizó el diseño e implementación del programa, ya que un ente ajeno al programa podrá darle un aspecto crítico más objetivo que el diseñador del programa. Este método trata de verificar si el programa no deja muchos huecos donde el usuario pueda entrar "accidentalmente" y dañar la ejecución del mismo. Para el éxito de esta prueba, el software debe obviar estas acciones extraordinarias o avisar al usuario sobre la imposibilidad de realizar esta acción.

Capítulo 1: Fundamentación Teórica

1.7 Estimación en prueba.

La estimación del coste y del esfuerzo del software nunca será una ciencia exacta. Son demasiadas las variables humanas, técnicas, de entorno y políticas que pueden afectar el coste final del software y al esfuerzo aplicado para desarrollarlo. Sin embargo, la estimación del producto de software puede dejar de ser un oscuro arte para convertirse en una serie de pasos sistemáticos que proporcionen estimaciones con un grado de riesgo aceptable. (Pressman, 2005)

El jefe del proyecto debe establecer expectativas sobre el tiempo necesario para completar el software entre las partes interesadas, el equipo, y la organización de la gestión, si esas expectativas no son realistas a partir del comienzo del proyecto, los interesados no confían en el equipo o en el director del proyecto.

Para generar una buena estimación, un proyecto debe tener: (Stellman, 2005)

- Una estructura de desglose de trabajo (WBS), o una lista de tareas que, de ser completada, producirá el último producto.
- Una estimación del esfuerzo para cada tarea.
- Una lista de supuestos que eran necesarios para hacer la estimación.
- Consenso con el equipo del proyecto para que las estimaciones sean más precisas.

1.7.1 Técnicas de estimación del proceso de desarrollo de software.

Estas técnicas de estimación son una forma de resolución de problemas en donde, en la mayoría de los casos, el problema a resolver es demasiado complejo para considerarlo como una sola parte. Por esta razón, descomponemos el problema, recaracterizándolo como un conjunto de pequeños problemas. (Pressman, 2005)

Líneas de Código y Puntos de Función.

Los datos de líneas de código (LDC) y los puntos de función (PF) se emplean de dos formas durante la estimación del proyecto de software:

- Variables de estimación, utilizadas para calibrar cada elemento del software.
- Métricas de base, recogidas de anteriores proyectos utilizadas junto con las variables de estimación para desarrollar proyecciones de costo y esfuerzo.

Capítulo 1: Fundamentación Teórica

Las líneas de código y los puntos de función no son métodos eficientes para estimar las pruebas en los proyectos de la universidad puesto que estos son grandes, orientados a objetos o que cuentan con documentación asociada, estos métodos eran eficientes cuando se usaba programación estructurada.

Técnicas Delphi.

Las técnicas delphi fueron desarrolladas en la corporación Rand en el año 1948, con el fin de obtener el consenso de un grupo de expertos sin contar con los efectos negativos de las reuniones de grupos. La técnica puede adaptarse a la estimación de costos de la siguiente manera: (Stellman, 2005)

- Un coordinador proporciona a cada experto la documentación con la definición del sistema y una papeleta para que escriba su estimación.
- Cada experto estudia la definición y determina su estimación en forma anónima; los expertos pueden consultar con el coordinador, pero no entre ellos.
- El coordinador prepara y distribuye un resumen de las estimaciones efectuadas, incluyendo cualquier razonamiento extraño efectuado por alguno de los expertos.
- Los expertos realizan una segunda ronda de estimaciones, otra vez anónimamente, utilizando los resultados de la estimación anterior. En los casos que una estimación difiera mucho de las demás, se podrá solicitar que también en forma anónima el experto justifique su estimación.
- El proceso se repite varias veces como se juzgue necesario, impidiendo una discusión grupal durante el proceso.
- El siguiente enfoque es una variación de la técnica Delphi tradicional que aumenta la comunicación conservando el anonimato.
- El coordinador proporciona a cada experto la documentación con la definición del sistema y una papeleta para que escriba su estimación.
- Cada experto estudia su definición, y el coordinador llama a una reunión del grupo con el fin de que los expertos puedan analizar los aspectos de la estimación con él y entre ellos.
- Los expertos terminan su estimación en forma anónima.
- El coordinador prepara un resumen de las estimaciones efectuadas sin incluir los razonamientos realizados por algunos de los expertos.
- El coordinador solicita una reunión del grupo para discutir los puntos donde las estimaciones varíen más.

Capítulo 1: Fundamentación Teórica

- Los expertos efectúan una segunda ronda de estimaciones, otra vez en forma anónima. El proceso se repite tantas veces como se juzgue necesario.

Delphi es un buen método de estimación pero necesita de especialistas o expertos altamente capacitados sobre el proyecto al cual se le van a realizar las pruebas.

COCOMO.

El Modelo Constructivo de Costos (COnstructive COst MOdel) es una jerarquía de modelos de estimación para el software. Esta jerarquía está constituida por los siguientes modelos:

- El modelo COCOMO intermedio: calcula el esfuerzo del desarrollo de software en función del tamaño del programa y de un conjunto de conductores de costo, que incluyen la evaluación subjetiva del producto, del hardware, del personal y de los atributos del proyecto.
- El modelo COCOMO avanzado: incorpora todas las características de la versión intermedia y lleva a cabo una evaluación de impacto de los conductores de costo en cada fase (análisis, diseño, entre otros) del proceso de ingeniería de software.

Se han desarrollado varias técnicas de estimación para el desarrollo de software, como establecer de antemano el ámbito del proyecto, usar las métricas del software (mediciones del pasado) como base para la realización de estimaciones y desglosar el proyecto en partes más pequeñas que se estiman individualmente. Esto ayuda al programador, ya que le permite dedicar más tiempo a otras partes del proyecto. Este método es muy utilizado pero debido a las características específicas con las que cuenta la universidad no sirve de gran ayuda para la estimación de las pruebas, pues se basan en entradas y salidas y en la universidad los proyectos generalmente utilizan metodología RUP.

Puntos de Caso de Uso.

El método de estimación puntos de casos de uso es el antecedente para el método de estimación de pruebas de liberación en el desarrollo del software en la universidad, ya que el mismo asigna pesos a un cierto número de factores que le afectan, para finalmente contabilizar el tiempo total estimado para el proyecto a partir de dichos factores.

Factor de Peso de los Actores sin ajustar (UAW): Este valor se calcula mediante un análisis de la cantidad de actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los actores

Capítulo 1: Fundamentación Teórica

se establece teniendo en cuenta en primer lugar si se trata de una persona o de otro sistema, y en segundo lugar, la forma en la que el actor interactúa con el sistema.

Los criterios se muestran en la **Tabla 1**.

Tipo de Actor	Descripción	Factor de Peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, Application Programming Interface)	1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3

Tabla 1: Clasificación de la experiencia de los actores.

Factor de Peso de los Casos de Uso sin ajustar (UUCW): Este valor se calcula mediante un análisis de la cantidad de casos de uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los casos de uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una transacción se entiende como una secuencia de actividades atómica, es decir, se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia y está representada por uno o más pasos del flujo de eventos principal del caso de uso, pudiendo existir más de una transacción dentro del mismo caso de uso. Los criterios se muestran en la **Tabla 2**.

Tipo de Caso de Uso	Descripción	Factor de Peso
Simple	El caso de uso contiene de 1 a 3 transacciones.	5
Medio	El caso de uso contiene de 4 a 7 transacciones.	10
Complejo	El caso de uso contiene más de 8 transacciones.	15

Capítulo 1: Fundamentación Teórica

Tabla 2: Clasificación de los casos de usos.

1.8 Pruebas en la UCI.

La UCI como institución productora de software vela por la calidad de sus productos, para lo cual el software pasa por un proceso de pruebas desde sus comienzos y antes de ser entregados al cliente como estrategia de la UCI se le desarrolla el proceso de pruebas de liberación. (Calisoft, 2008)

Para realizar las pruebas de liberación a un proyecto, es necesario que el proyecto haga una solicitud formal a la Dirección de Calidad Central, la cual es realizada generalmente por parte del jefe de proyecto que quiere liberar, donde explica detalladamente todos los módulos de los artefactos que quieren liberar, dígase documentación, bases de datos, documento de arquitectura o la aplicación como tal. La Dirección de Calidad hace la planificación y decide en que fecha comienzan a liberar, se hace una reunión de inicio donde se entrega toda la documentación necesaria para las pruebas: manual de usuario, manual de instalación, diccionario de datos, roles y usuario, la descripción de los casos de usos, la especificación de los requisitos funcionales y la aplicación que se va a probar, en dicha reunión se define los tipos de pruebas que se le realizarán al software. Luego se distribuye el trabajo a los estudiantes del laboratorio industrial de pruebas de software (LIPS) perteneciente a la Dirección de Calidad, con el objetivo de que por cada caso de uso diseñe un caso de prueba, lo ideal sería que cada producto ya traiga diseñado los casos de prueba, en este caso también se hace la distribución para una revisión de los casos de prueba, y partiendo de ahí se comenzará a probar.

En caso de que el LIPS no cuente con la cantidad de estudiantes necesarios para realizar las pruebas, se utilizarán los estudiantes pertenecientes al proyecto de calidad existente en cada facultad, guiados por un asesor de calidad del mismo, y al frente de ellos se encontrará un especialista de la Dirección de Calidad, esto mayormente ocurre en el cambio de semestre cuando los estudiantes del LIPS se encuentran en exámenes. **Ver Anexo 2.**

Capítulo 1: Fundamentación Teórica

1.9 Propuesta de solución.

La propuesta de solución consiste en la elaboración de un procedimiento para estimar tiempo y esfuerzo de las pruebas de liberación, donde se tendrán en cuenta: la complejidad de los casos de uso, experiencia de los probadores y riesgos asociados. El procedimiento tiene como antecedente al método de estimación basado en puntos de casos de uso.

Para planificar la estimación se tendrán en cuenta dos etapas: el diseño y la implementación de las pruebas, en el diseño las entradas para llevar a cabo el procedimiento son: especificación de requisitos, casos de uso y procesos, donde se tendrán en cuenta los siguientes puntos, para a partir de ellos estimar el tiempo que demora el diseño de las pruebas y el esfuerzo que se necesita, estos son: complejidad de los artefactos, para esta complejidad se analizará la cantidad de escenarios y el volumen de datos de entrada y salida teniendo presente la criticidad de los casos de uso, para el negocio-cliente, y para la arquitectura, otro punto a tener en cuenta en el diseño es la experiencia del diseñador y los riesgos. En la implementación, la entrada a las pruebas serán: los casos de prueba, los manuales y las listas de chequeo, analizándose también la experiencia y los riesgos además de la complejidad de los casos de usos, aquí aparece un nuevo factor que es la utilización de herramientas automatizadas que pueden ayudar a disminuir el tiempo de desarrollo de las pruebas.

Estimación (Tiempo y esfuerzo)

➤ Diseño

Entradas: especificación de requisitos, casos de uso, procesos

- Complejidad de los artefactos.
 - ✓ Cantidad de Escenarios.
 - ✓ Volumen de Datos de entrada y salida.
 - ✓ Criticidad de los casos de uso.
 - Para el negocio - cliente.
 - Para la arquitectura.
- Experiencia del diseñador.
- Riesgos.

➤ Implementación de pruebas

Entradas: caso de prueba, manuales, listas de chequeo.

Capítulo 1: Fundamentación Teórica

- Riesgos.
- Experiencia.
- Complejidad del caso de prueba.

Uso de herramientas automatizadas

1.10 Conclusiones Parciales

En el capítulo se introdujeron conceptos con el fin de lograr un mejor entendimiento del proceso de estimación, específicamente en pruebas de liberación. Se abarcó acerca de los niveles, tipos y estrategias de las pruebas. Además se realizó un estudio comparativo entre los diferentes métodos de estimación, haciendo énfasis en el método basado en puntos de casos de uso.

Capítulo 2: Propuesta de Solución

Capítulo II: Propuesta de solución.

2.1 Introducción.

Después de analizar los diferentes métodos de estimación de las pruebas en el desarrollo del software y en especial el método de puntos de casos de usos, se propone un procedimiento para estimar tiempo y esfuerzo de las pruebas de liberación en el laboratorio de Calidad Central, debido a que en este laboratorio no existe ningún método o procedimiento de estimación de las pruebas de liberación, sino que se basan en la experiencia de los especialistas.

2.2 Propuesta del procedimiento.

El procedimiento de estimación de tiempo y esfuerzo en pruebas de liberación será aplicado en el Laboratorio Industrial de Prueba de Software (LIPS), perteneciente al Laboratorio de Pruebas de la Dirección de Calidad de Software. Se estimará el esfuerzo y tiempo empleado en el diseño e implementación de las mismas.

Para obtener los resultados óptimos del procedimiento es necesario tener en cuenta las siguientes restricciones:

- Solo se utilizará en proyectos que utilicen metodología RUP: La metodología RUP tiene tres características fundamentales: centrado en la arquitectura, iterativo e incremental y guiado por casos de uso, esta última constituye el principal artefacto de esta metodología y es una de las formas más completas de especificar los requisitos del sistema, a través de ellos se sigue la traza de todo el sistema. Estos dicen al cliente que esperar (a través de la descripción textual se especifica la interacción cliente-sistema), al diseñador que clase diseñar, para luego poder implementar, al revisor técnico de que trata el documento, y al probador que debe probar.

En una encuesta realizada a los proyectos productivos de la Universidad de las Ciencias Informáticas en el año 2008 se arrojó como resultado que la metodología más utilizada en la universidad es la metodología RUP, con un 65 %. Debido a que la gran mayoría de los proyectos que pasarán por el laboratorio de pruebas utilizarán dicha metodología, se tomó en consideración realizar este procedimiento para proyectos que utilicen metodología RUP.

Capítulo 2: Propuesta de Solución

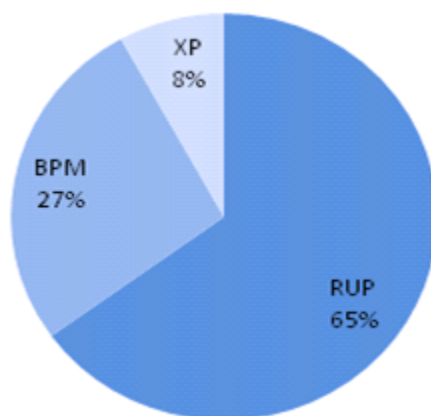


Gráfico 1: Metodologías aplicadas en los proyectos de la UCI.

- Se asignará un grupo de probadores que participarán en todo el desarrollo de las pruebas, proporcionando la familiarización de los probadores con el negocio.
- En caso que exista mayor cantidad de probadores que casos de uso, el número de probadores que se utilizará en la iteración, será igual al número de casos de uso. Ya que se ha demostrado que no cumple objetivo poner dos o más probadores a diseñar e implementar un caso de prueba, pues consumen el mismo tiempo de un solo probador.

Pasos a seguir:

1. Identificar el proyecto a probar.

Se hace una solicitud forma al laboratorio de Calidad Central, realizada generalmente por parte del jefe de proyecto que desee liberar, esta solicitud se realiza mediante el llenado de una plantilla (“Solicitud de Servicios de Liberación de Productos del Desarrollo de Software”), donde explica detalladamente todos los módulos y artefactos a probar, dígase documentación, bases de datos, documento de arquitectura o la aplicación como tal. Cuando el jefe del laboratorio recibe dicha solicitud, asigna un especialista del laboratorio de prueba, que se desempeñará como coordinador de prueba y será el intermediario entre el equipo de desarrollo y el equipo de prueba.

Capítulo 2: Propuesta de Solución

Este especialista es el encargado de definir las pruebas que se le deben realizar a la aplicación, un tipo de prueba que es obligatorio realizar son las pruebas funcionales, las cuales demuestran la correctitud de los requisitos funcionales del sistema. En el caso de que los proyectos se desarrollen utilizando metodología RUP, los casos de pruebas se diseñan basados en los casos de uso.

El diseño de los casos de prueba se realiza a través de una plantilla (Diseño de Casos de Pruebas Funcionales basados en casos de uso), la cual consta de una serie de elementos de estricto cumplimiento: (Calisoft, 2008)

- ✓ Descripción General.
- ✓ Condiciones de Ejecución.
- ✓ Secciones a probar en el Caso de Uso.
- ✓ Descripción de Variables.
- ✓ Matriz de Datos.
- ✓ Registro de Defectos y Dificultades Detectadas.

Luego se realiza una reunión de inicio coordinada por el especialista del laboratorio de pruebas de liberación, donde dicho laboratorio recibe toda la documentación necesaria para las pruebas: manual de usuario, manual de instalación, diccionario de datos, roles y usuarios, descripción de los casos de uso, especificación de los requisitos funcionales y la aplicación que se va a probar, en esta reunión también se determina la presencia de un especialista por parte del proyecto que desea liberar, que junto al especialista del laboratorio de pruebas de liberación se encargarán de llevar a cabo el proceso de pruebas de liberación. Este especialista debe tener un amplio conocimiento de las características propias del negocio y del funcionamiento de la aplicación.

2. Asignar probadores.

Se escogerá dentro del grupo de probadores del laboratorio de pruebas de liberación a aquellos probadores que sean necesarios para realizar las pruebas al producto que se desee liberar. Estos probadores deben estar capacitados sobre el negocio del producto que se va a liberar, manteniéndose durante todo el ciclo del desarrollo de las pruebas, con el objetivo de lograr una estabilidad de los

Capítulo 2: Propuesta de Solución

probadores, teniéndose en cuenta que los estudiantes que van a probar serán aquellos que cuenten con menor carga docente, teniendo como primera prioridad el proceso de pruebas de liberación.

3. Clasificación de los elementos a tener en cuenta en el procedimiento.

Para obtener información sobre el proceso de diseño e implementación de las pruebas de liberación en el laboratorio de Calidad Central se puso en práctica el modelo “Tormenta de Ideas”, a un grupo de especialistas del mismo, con el objetivo de potenciar su participación y creatividad, enfocándolos hacia un objetivo común, mediante este sistema se trata primero de generar las ideas y luego evaluarlas.

Tormenta de Ideas.

Es una reunión o dinámica de grupo que emplea un moderador y un procedimiento para favorecer la generación de ideas. La producción de ideas en grupo puede ser más efectiva que individualmente. (Bartle, 2009)

➤ **Fundamento del modelo:**

- ✓ Muchas ideas mueren por la crítica destructiva a que se ven sometidas antes de que maduren o se perfeccionen. Mediante este sistema se trata primero de generar las ideas y luego de evaluarlas.
- ✓ La reunión trata de crear un clima distendido que favorezca la comunicación y la participación de los asistentes. Crear un ambiente que favorezca la comunicación y la libre exposición de las ideas es fundamental. La motivación de los miembros de grupo es imprescindible, por lo que la reunión debe resultar relajada, amena e incluso divertida.
- ✓ Por tanto, es fundamental el entorno en el que se desarrolla la sesión y el crear un clima que facilite la exposición de ideas sin cortapisas. Se fomenta la participación de todos los miembros del equipo y en un principio las ideas de los demás no se critican por muy descabelladas que puedan parecer.
- ✓ Se trata de poner en juego la imaginación y la memoria de forma que una idea lleve a la otra. El modelo trata de fomentar las asociaciones de ideas por semejanzas o por oposición.

Capítulo 2: Propuesta de Solución

➤ **Características principales**

A continuación se comentan una serie de características que ayudan a comprender la naturaleza del modelo:

- ✓ **Participación:** Favorecen la intervención múltiple de los participantes, enfocándola hacia un tema específico, de forma estructurada y sistemática.
- ✓ **Creatividad:** Las reglas a seguir para su realización favorecen la obtención de ideas innovadoras. Estas son en general, variaciones, reordenaciones o asociaciones de conceptos e ideas ya existentes.

➤ **Propósito.**

El propósito de una sesión de tormenta de ideas es trabajar como grupo para identificar un problema, y hallar, a través de una intervención participativa, la mejor decisión de grupo para un plan de acción que lo solucione.

➤ **Presentaciones.**

La tormenta de ideas requiere una preparación previa, se debe disponer de un lugar en el que se pueda dialogar sin interrupciones y de forma relajada. Una sala en la que los miembros puedan debatir sin distracciones. Lo ideal es contar con una pizarra a la vista de todos y elementos para escribir. También puede utilizarse un panel donde se van colocando a la vista de todos, las ideas que se van generando y las posteriores relaciones entre las mismas. Se convoca a los miembros, estableciendo la duración de la misma, el problema que se trata de solucionar, la mecánica que se utiliza y la cuestión en la que se centrará la creatividad.

➤ **Fases.**

La reunión consta de dos fases:

- ✓ En la primera fase todos aportan ideas, pero no se permite ninguna crítica o juicio sobre estas. A partir de las ideas iniciales propuestas por los distintos miembros del grupo, se van generando nuevas rondas de ideas o ideas derivadas. En la primera fase se trata de producir un gran número de ideas aunque parezcan inútiles o descabelladas.

Capítulo 2: Propuesta de Solución

- ✓ En la segunda fase y dirigidos por el moderador, las ideas se seleccionan y se realiza un examen crítico. La selección de ideas puede dejarse para una segunda reunión o incluso es preferible que sea otro grupo el que seleccione y enjuicie de modo crítico las ideas.

➤ **Etapas.**

- ✓ **Pre calentamiento:** Se comienza por crear un clima relajado tratando durante unos minutos un tema sencillo y no comprometido. Importante cuando existen miembros sin experiencia en la técnica con ciertas inhibiciones para expresarse.
- ✓ **El moderador plantea el problema:** Utiliza preguntas como ¿Por qué? ¿Cómo? Se determina el problema, precisándolo y delimitándolo, brevemente se expone el punto de partida, la situación actual y las experiencias que se poseen.
- ✓ Los componentes reflexionan y escriben cada uno una amplia lista de soluciones o alternativas.
- ✓ Cada miembro expone sus soluciones en alto sin debatirlas. No se permite en esta fase rebatir o enjuiciar las alternativas de los demás.
- ✓ Una vez terminada la primera ronda de exposiciones el grupo partiendo de las primeras ideas de forma conjunta va proponiendo nuevas soluciones o alternativas. Las ideas se anotan en un lugar visible. Las imágenes y los gráficos visuales ayudan a fomentar la creatividad y a definir posteriormente las ideas.
- ✓ Se analiza la forma de combinar las distintas alternativas y se generan nuevas ideas. Se trata de analizar como pueden relacionarse las ideas anteriormente dispersas. Se agrupan las ideas y se relacionan.
- ✓ Se enumeran todas las ideas seleccionadas.
- ✓ Una etapa posterior que puede realizar el mismo grupo o un grupo distinto es la evaluación de las ideas seleccionadas. Se trata de descubrir soluciones. Se seleccionan las ideas más útiles y si es necesario se ponderan.
- ✓ **Enriquecimiento:** Hay que definir con detalle la idea, mediante un esquema o dibujo.

➤ **Requerimientos:**

- ✓ Un problema que solucionar.

Capítulo 2: Propuesta de Solución

- ✓ Un grupo con potencial para trabajar en equipo. Puede ser desde un pequeño equipo operacional o gestor de entre cinco y diez personas, hasta una reunión de todo un pueblo de cien o doscientas personas.
- ✓ Un tablero, grandes hojas de papel en blanco o algo que sea fácilmente visible por todos, y algunos rotuladores grandes para escribir.
- ✓ Un moderador, alguien cuya función es provocar las sugerencias de los participantes, no imponerles sus opiniones, aunque con aptitudes de liderazgo para mantener el orden y el propósito de la sesión.

➤ **Las reglas básicas:**

- ✓ El moderador dirige cada sesión.
- ✓ El moderador pide sugerencias de los participantes.
- ✓ No se permite la crítica a las sugerencias de cualquiera, por parte de nadie.
- ✓ Todas las sugerencias se registran en la pizarra, incluso las disparatadas.

➤ **El equipo de la tormenta de ideas**

Los miembros del grupo deben encontrarse motivados y relajados para exponer ideas sin cortapisas, ni prejuicios. Se debe generar una cultura de innovación que no se limite a las ideas preconcebidas. Hay que seguir cuatro reglas básicas:

- ✓ Ninguna crítica: Es preciso evitar las ideas preconcebidas y las expresiones que puedan bloquear la generación y expresión libre de las ideas. Evitar los bloqueos que los hábitos, los procedimientos establecidos, la cultura y las normas ponen a la generación de ideas.
- ✓ No ser convencional: No todo está inventado. Los procedimientos actuales pueden cambiar por otros que requieran menos tiempo, menores gastos, menos recursos o que contaminen menos. No dar nada por sentado.
- ✓ Cuantas más ideas mejor.
- ✓ Apoyarse en otras ideas: Aprender a partir de unas ideas para llegar a otras. Trabajar con ideas visuales

Capítulo 2: Propuesta de Solución

➤ Procedimiento:

1. Definir el problema:

- Pida sugerencias de cuál es el problema más importante.
- No permita las críticas a las sugerencias de los demás, por parte de nadie.
- Escriba en la pizarra todos los problemas propuestos.
- Agrupe los problemas similares o relacionados.
- Ordénelos y lístelos por orden de prioridad, empezando por los más importantes.

2. Señalar la meta:

- Invierta la definición del problema (su solución).
- La solución al problema definido en la etapa anterior es la meta.
- Defina la meta como solución al problema.
- Escriba la meta en la pizarra.
- Recuerde al grupo que esta meta es la que han elegido.

3. Definir el objetivo:

- Explique la diferencia entre una meta y un objetivo.
 1. El moderador debe saber que un objetivo debe ser medible, finito y tener una fecha de conclusión.
 - **Objetivo:** “Objetivo no es más que la expresión de un deseo mediante acciones concretas para conseguirlo”. (Bartle, 2009)
Por ello un objetivo sirve para:
 - Formular concreta y objetivamente resultados
 - Planificar acciones
 - Orientar procesos
 - Medir resultados
 - **Meta:** La meta se define fácilmente como la solución del problema que se ha identificado. El inconveniente es que una meta es demasiado general; no es fácil obtener un consenso sobre cuándo se ha logrado. Un objetivo se deriva de una

Capítulo 2: Propuesta de Solución

meta, tiene la misma intención que una meta, pero es más específico, cuantificable y verificable que la meta.

- Pida al grupo que sugiera objetivos.

Los objetivos deben ser:

- ✓ Específicos: claros sobre qué, dónde, cuándo y cómo va a cambiar la situación.
- ✓ Medibles: que sea posible cuantificar los fines y beneficios.
- ✓ Realizables: que sea posible lograr los objetivos (conociendo los recursos y las capacidades a disposición de la comunidad).
- ✓ Realistas: que sea posible obtener el nivel de cambio reflejado en el objetivo.
- ✓ Limitado en tiempo: estableciendo el periodo de tiempo en el que se debe completar cada uno de ellos.

- Escriba todos los objetivos propuestos en la pizarra.
- No acepte críticas a ninguna sugerencia, por parte de nadie.
- Agrupe los objetivos similares o relacionados.
- Ordénelos y lístelos por orden de prioridad, empezando por los más importantes.
- Recuerde al grupo que son ellos los que han generado los objetivos principales.

4. Identificar recursos e impedimentos:

- Pida al grupo que sugiera recursos e impedimentos.
- Escriba todos los recursos e impedimentos en la pizarra.
- No acepte críticas a ninguna sugerencia, por parte de nadie.
- Agrupe los recursos similares o relacionados.
- Ordénelos y lístelos por orden de prioridad, empezando por los más importantes.
- Recuerde al grupo que son ellos, no usted, los que han generado la lista.
- Agrupe los impedimentos similares o relacionados.
- Ordénelos y lístelos por orden de prioridad, empezando por los más importantes.
- Recuerde al grupo que son ellos los que han generado el orden de la lista.

5. Identificar una estrategia:

- Pida al grupo que sugiera estrategias.

Capítulo 2: Propuesta de Solución

Estrategia: Es el proceso a través del cual una organización formula objetivos, y está dirigida a la obtención de los mismos. Estrategia es el medio, la vía, es el cómo para la obtención de los objetivos de la organización. Es el arte (maña) de entremezclar el análisis interno y la sabiduría utilizada por los dirigentes para crear valores de los recursos y habilidades que ellos controlan. Para diseñar una estrategia exitosa hay dos claves; hacer lo que hago bien y escoger los competidores que puedo derrotar. Análisis y acción están integrados en la dirección estratégica". (Halten, 1987)

La estrategia se basa en cinco atributos:

- ✓ Se ocupa de las cuestiones fundamentales.
 - ✓ Ofrece un marco de referencia para una planeación más detallada y para las decisiones ordinarias.
 - ✓ Supone un marco temporal más largo.
 - ✓ Ayuda a orientar las energías y recursos de la organización hacia las actividades de alta prioridad.
 - ✓ Es una actividad de alto nivel, en el sentido de que la alta gerencia debe participar.
-
- Escriba las estrategias propuestas en la pizarra.
 - No acepte críticas a ninguna sugerencia, por parte de nadie.
 - Agrupe las estrategias similares o relacionadas.
 - Ordénelas y lístelas por orden de prioridad, empezando por las más importantes.
 - Recuerde al grupo que son ellos los que han generado la lista.
 - Elija la estrategia situada al principio de la lista.

6. Resumir en la pizarra las decisiones del grupo:

- el problema
- la meta
- los objetivos
- los recursos

Capítulo 2: Propuesta de Solución

- los impedimentos
- la estrategia

Informe al grupo de que han producido un plan de acción. Si alguno se presta a escribir lo que se ha decidido en cada una de las categorías de arriba, tendrán el núcleo de un documento estándar de planificación. Hágalos saber que lo han elaborado como equipo, y que el plan les pertenece.

El modelo Tormenta de ideas es simple, lo que no significa que sea fácil, mejora con la práctica. Puede intercalar las diferentes fases del proceso con juegos de rol, juegos de grupo y otras técnicas de trabajo en equipo.

La tormenta de ideas se aplicó a los expertos del grupo de diseño de los casos de prueba del laboratorio de pruebas de liberación de la Dirección de Calidad, este grupo está conformado por siete expertos, dedicados al diseño de los casos de prueba desde que surgió Calidad Central. Este modelo se aplicó para obtener informaciones, datos y clasificaciones de los elementos que se tuvieron en cuenta en el procedimiento.

Después de la realización del modelo “Tormenta de Ideas “se llegó a la conclusión que para la estimación de tiempo y esfuerzo del diseño e implementación de las pruebas de liberación hay dos elementos fundamentales a tener en cuenta, ellos son:

- ✓ Experiencia de los probadores.
- ✓ Complejidad de los casos de uso.

Se determinó como factores importantes la experiencia de los probadores, debido a que demuestra cuan certero puede ser el diseño e implementación de los casos de prueba, y la complejidad de los casos de uso, que miden el grado de dificultad al que se va a enfrentar un probador. Estos dos elementos son la base para un óptimo diseño e implementación de los casos de prueba.

➤ **Clasificación de la experiencia de los probadores.**

La experiencia de los probadores se clasificó en:

Capítulo 2: Propuesta de Solución

- ✓ Probadores de experiencia alta.
- ✓ Probadores de experiencia media.
- ✓ Probadores de experiencia baja.

Para ello los expertos se basaron en la gestión por competencia, la cual medirá una serie de elementos que se incluirán dentro de conocimientos y habilidades, referidos a los conocimientos alcanzados por un probador en asignaturas como: Ingeniería de Software, Gestión de Software y Programación y las habilidades alcanzadas en herramientas automatizadas.

Gestión por competencia.

Es el proceso que consiste en administrar el activo intelectual e intangible que representan las competencias en las personas. La competencia integra dos dimensiones: (Domingo J, 2000)

- ✓ Características personales y sociales (aptitudes emocionales sobre las que se conforma la inteligencia emocional).
- ✓ Conocimientos precisos para la realización adecuada de las actividades y habilidades técnicas contextualizadas a los procesos de trabajo.

➤ **Tipos de competencias:**

- ✓ Las competencias diferenciadoras: Distinguen un trabajador con actuación superior de un trabajador con actuación mediana.
- ✓ Las competencias umbral o esenciales: Son las necesarias para lograr una actuación media o mínimamente adecuada.

Todas las personas poseen un conjunto de atributos y conocimientos que pueden ser tanto adquiridos como innatos, y que definen sus competencias para una determinada actividad. Las competencias claves tienen una influencia decisiva en el desarrollo del puesto de trabajo y por tanto en el funcionamiento de la empresa. Si se produce una adecuación (un ajuste) entre las competencias de la persona y las requeridas por su puesto, el ajuste será óptimo. Si por el contrario, hay un desfase entre las competencias requeridas por el puesto y las características de la persona, su adecuación se verá afectada negativamente.

Capítulo 2: Propuesta de Solución

➤ Clasificación de las competencias:

De acuerdo al grado de especificidad, las competencias se dividen en dos grandes bloques:

- ✓ Conocimientos específicos, de carácter técnico, precisos para la realización adecuada de las actividades.
- ✓ Habilidades/cualidades: Capacidades específicas precisas para garantizar el éxito en el puesto.

En la **Tabla 3** se especifican detalladamente los elementos tenidos en cuenta en la gestión por competencia, para determinar la experiencia de los probadores y el peso que se le otorga a cada una, el cual es inversamente proporcional a la clasificación de la experiencia, es decir, un probador de experiencia baja necesita un mayor peso, puesto que demorará mayor tiempo en diseñar o implementar un caso de prueba. Para seleccionar la categoría de la experiencia de un probador solo se puede incumplir con una de las habilidades o conocimientos establecidos en la categoría de cada clasificación. En el caso que un probador no encuentre clasificada su experiencia en alta o media, se le otorgará la clasificación de experiencia baja, puesto que no cumple con las condiciones requeridas expuestas anteriormente.

Experiencia	Conocimientos	Habilidades	Peso
Alta	-Del negocio del sistema a probar. - Avanzados en ISW. -Sobre otras metodologías de desarrollo de software.	-Haber participado en pruebas en proyectos similares. -Lógica necesaria para ejecutar pruebas técnicas.	0.5
Media	-Tiene que haber vencido ISW. -Tener acreditado cursos optativos relacionados con ISW y Calidad.	-Haber participado en al menos una iteración de pruebas.	1.0
Baja			1.5

Tabla 3: Clasificación de los probadores teniendo en cuenta su experiencia.

Capítulo 2: Propuesta de Solución

En el **Anexo 3** se muestra la aplicación del modelo "Tormenta de ideas", el cual dio como resultado: los elementos que se tendrían en cuenta para clasificar la complejidad de los casos de uso, basándose para ello en datos históricos tomados de los seis mayores proyectos que se han liberado en el centro (Prisiones, Registros y Notarías, Identidad, Aduana, CICPC, SIGESC). Ver **Anexo 4**. En la **Tabla 4** se muestran los datos más generales de los proyectos.

Proyectos	Cantidad de Casos de Uso	Cantidad de Sesiones	Cantidad de Flujos Básicos	Cantidad de Flujos Alternativos	Cantidad de Elementos de E/S
Aduana	14	14	14	7	644
CICPC	18	47	18	145	857
Identidad	13	24	13	40	112
Prisiones	12	12	12	12	255
SIGESC	221	318	221	652	2794
Registros y Notarías	26	52	26	69	731

Tabla 4: Elementos representativos de los seis mayores proyectos.

➤ **Clasificación de los casos de uso:**

Los casos de uso se clasifican teniendo en cuenta:

- ✓ Cantidad de Flujos Alternativos.
- ✓ Cantidad de elementos de entrada y/o salida (elementos E/S).

Estos elementos fueron los que se escogieron para clasificar los casos de uso debido a que arrojaron los valores más significativos después del estudio realizado a los datos históricos.

En la **Tabla 5** se especifican los elementos tenidos en cuenta para clasificar la complejidad de los casos de uso, así como su clasificación y el peso que se le otorga a cada uno. En este caso el peso es directamente proporcional a la clasificación de la complejidad de los casos de uso porque entre mayor sea su clasificación mayor será el grado de dificultad para desarrollar dicho caso de uso.

Capítulo 2: Propuesta de Solución

Clasificación	Elementos de E/S	Cantidad de flujos	Peso
Simple	1 – 6	1 – 3	2
Medio	7 – 15	4 – 7	3
Complejo	Mayores que 15	Mayores que 7	4

Tabla 5: Clasificación de los casos de usos teniendo en cuenta su complejidad.

Un caso de uso se clasifica de acuerdo a la cantidad de elementos de E/S y después de acuerdo a la cantidad de flujos alternativos con las que cuente el caso de uso, la mayor de estas clasificaciones será la clasificación final del caso de uso, la cual será a la que se le otorgue el peso.

Para clasificar la complejidad de un caso de uso es necesario:

- Determinar la cantidad de sesiones por casos de uso.
Para determinar la cantidad de sesiones por casos de uso es necesario analizar la descripción de cada caso de uso, si el caso de uso cuenta con más de una sesión (flujos excluyentes dentro de un caso de uso), cada sesión se va a tomar como un caso de uso independiente.
- Determinar la cantidad de elementos de E/S por casos de uso.
Para determinar la cantidad de elementos de E/S por casos de uso es necesario analizar la descripción de cada caso de uso y contar dichos elementos.
- Determinar la cantidad de flujos alternativos por casos de uso.
Para determinar la cantidad de flujos alternativos por casos de uso es necesario analizar la descripción de cada caso de uso y contar los flujos alternativos.

4. Distribución de casos de uso por probadores.

Generalmente el planificador es quien estima el tiempo de duración de las pruebas y por tanto quien distribuye los casos de uso por probador. Para que la distribución sea de una manera mas rápida se sugiere que se empiece a distribuir por la mayor de las complejidad de los casos de uso hacia la menor de ellas y de igual manera los probadores deben estar organizados de mayor a menor experiencia, para tratar de hacer coincidir la mayor de las veces o tanto como sea posible un probador con una elevada experiencia con aquel caso de uso que tenga la mayor complejidad en ese momento.

Capítulo 2: Propuesta de Solución

Se le asigna a los probadores los casos de uso de manera tal que todos tengan un esfuerzo similar para diseñar un caso de prueba, es necesario apoyarse en la **Tabla 6** la cual muestra el esfuerzo que realiza un probador con una categoría determinada en diseñar un caso de prueba para un caso de uso, basándose en las fórmulas de estimación de esfuerzo de diseño de caso de prueba.

Probador * Caso de Uso	Simple	Medio	Complejo
Baja	3.0	4.5	6.0
Media	2.0	3.0	4.0
Alta	1.0	1.5	2.0

Tabla 6: Esfuerzo empleado por un probador en diseñar un caso de prueba.

5. Seleccionar los casos de usos significativos.

Es necesario seleccionar los casos de uso significativos para poder estimar el tiempo de implementación de las pruebas exploratorias. Los directivos del laboratorio de pruebas tienen como acuerdo realizar una prueba exploratoria al software en la fase de implementación de las pruebas, para determinar cual es el nivel de calidad con el que cuenta.

Esta prueba se realiza con el objetivo de probar la funcionalidad de la aplicación, y verificar que no haya errores garrafales en la documentación, para eso se aplican criterios de criticidad, que establecen los parámetros para declarar un producto del desarrollo de software en estado crítico de terminación. Son aplicables en tres momentos fundamentales: cuando se hace la revisión de la solicitud (RS); durante el proceso de pruebas exploratorias (PE) y durante las pruebas formales (PF).

Los criterios de criticidad son los siguientes: (Calisoft, 2008)

- ✓ No se presenta la última versión del producto de trabajo comprobada y avalada por la etapa anterior.

Capítulo 2: Propuesta de Solución

- ✓ No están presentes todos los elementos componentes del sistema, producto o entregable. (Hardware, Software, Partes y Documentación). (RS) (PE)
- ✓ No se corresponden totalmente los requisitos funcionales documentados con los implementados. (RS) (PE)
- ✓ Supera el producto la tasa de: 1 defecto significativo por casos de uso para (PF) ó ½ defecto significativo por casos de uso para (PE) y en caso de otros artefactos al menos 2 defectos significativos que estén relacionados con las pautas esenciales definidas por la metodología que sigue el proyecto. (PF)
- ✓ Excede el producto el número máximo de iteraciones establecidas por plan (entre 2 y 3 iteraciones). (PF)
- ✓ Se incumple más del 50% de las reglas para la documentación establecidas por el proyecto y/o expediente del proyecto. (RS) (PE) (PF)
- ✓ Existe incoherencia significativa entre los artefactos que tienen relación o dependencia entre sí. (RS) (PE) (PF)
- ✓ Existe incoherencia significativa entre lo documentado y lo implementado. (RS) (PE) (PF)
- ✓ La cantidad de faltas de ortografía excede la cantidad de páginas o pantallas que tiene el artefacto en cuestión. (RS) (PE) (PF)
- ✓ Durante las pruebas de regresión persisten al menos 2 defectos significativos de iteraciones anteriores. (PE) (PF)
- ✓ Se observan los mismos tipos de defectos, ya señalados en iteraciones anteriores, en otros o los mismos lugares donde fueron detectados, para (PE): ½ de la cantidad de defectos tipo encontrados en etapa anterior, o para (PF): ¼ de la cantidad de defectos tipo encontrados en etapa anterior.

Capítulo 2: Propuesta de Solución

Si se aplica alguno de estos criterios de criticidad, se considera la prueba fallida y se para la prueba hasta el día siguiente como mínimo, o hasta que el proyecto este en condiciones de liberar. En este caso, el proyecto debe volver a hacer la solicitud cuando haya erradicado los señalamientos.

Un caso de uso es arquitectónicamente significativo si: (Franco, 2002)

- ✓ Contiene las principales funciones del negocio.
- ✓ Aporta las principales clases y componentes del sistema.
- ✓ El cliente quiere que ese caso de uso salga en la primera iteración.
- ✓ Implementando un caso de uso se pueden implementar otros que tengan un comportamiento parecido.

De manera resumida se puede decir que son aquellos que nos ayudan a mitigar los riesgos más importantes, deben ser los más importantes para los usuarios del sistema y que ayuden a cubrir las funcionalidades significativas, sobre todo y más importantes, los que dan solución a los requisitos no funcionales. Es necesario recalcar que la línea base de la arquitectura no debe quedar en análisis y diseño sobre papel, sino que debe completarse la iteración hasta implementación y prueba para lograr una implementación funcional que proporcione seguridad al arquitecto y a todos los miembros del equipo de desarrollo.

Procedimiento para estimar tiempo y esfuerzo de las pruebas de liberación.

- **Estimación de tiempo y esfuerzo para productos que requieran diseño e implementación de los casos de prueba.**

En el laboratorio de la Dirección de Calidad se pueden diseñar y/o implementar los casos de prueba, de acuerdo a las características propias del producto, si el producto no trae los casos de prueba diseñados, entonces los probadores del centro diseñarán e implementarán los casos de prueba. Lo ideal sería que cada producto cuente con los diseños de casos de prueba para solamente emplear tiempo en revisar estos diseños, y más tarde implementarlos.

Capítulo 2: Propuesta de Solución

Como resultados obtenidos del modelo “Tormenta de Ideas” se tiene:

- ✓ Siempre se estimará el tiempo y esfuerzo del diseño de los casos de prueba, ya que de esta estimación depende el tiempo de revisión e implementación de los casos de prueba.
- ✓ A la estimación del tiempo total se le sumará un riesgo, que será el 10% del tiempo de diseño más el tiempo de la implementación de los casos de prueba, este valor fue tomado en consideración con los tipos de riesgos a los cuales se enfrenta el proceso de pruebas de software.
- ✓ La estimación de tiempo y esfuerzo de la implementación consta de una prueba exploratoria y cuatro iteraciones, donde las dos primeras iteraciones serán 1/2 de la estimación del diseño, dado que se demorarán más diseñando casos de prueba que implementándolos. Las últimas dos iteraciones serán 1/3 de la estimación del diseño de los casos de prueba.

Pruebas Exploratorias.

La prueba exploratoria se define como el aprendizaje, el diseño de casos de prueba y la ejecución de las pruebas en forma simultánea. En otras palabras, es una técnica de prueba en la cual quien prueba controla activamente el diseño mientras son realizadas, y utiliza la información obtenida en la exploración para diseñar nuevas y mejores pruebas. (Wodzislowski Mónica, 2006)

- ✓ En la prueba exploratoria siempre se debe tomar nota de lo que se hizo y lo que sucedió.
- ✓ Los resultados de la prueba exploratoria no son necesariamente diferentes de aquellos obtenidos de la prueba con diseño previo y ambos enfoques para las pruebas son compatibles.
- ✓ La prueba exploratoria puede ser aplicada en cualquier situación donde no sea obvio cuál es la próxima prueba que se debe realizar.

La prueba exploratoria presenta una estructura externa fácil de describir. Durante un período de tiempo un probador interactúa con un producto para cumplir una misión y reportar los resultados. Una misión describe qué se probará del producto, los tipos de incidentes que se buscan y los riesgos involucrados. La misión puede ser elegida por el probador o serle asignada por el líder de pruebas.

Capítulo 2: Propuesta de Solución

❖ **Estimación de esfuerzo:**

$$E = ED + EI$$

$$ED = \sum_{i=1}^n ED_p$$

$$ED_p = PE_p * \sum_{i=1}^n PC_{CU}$$

$$EI = PE + \frac{1}{2}ED_1 + \frac{1}{2}ED_2 + \frac{1}{3}ED_3 + \frac{1}{3}ED_4$$

$$PE = 20\%ED$$

E: Esfuerzo total.

EI: Esfuerzo de la implementación de las pruebas.

ED: Esfuerzo del diseño de las pruebas.

ED_p: Esfuerzo del diseño de los probadores.

PE_p: Peso de la experiencia de los probadores.

PC_{CU}: Peso de la complejidad de los casos de uso.

PE: Prueba exploratoria.

ED₁ : Esfuerzo del diseño en la primera iteración.

ED₂ : Esfuerzo del diseño en la segunda iteración.

ED₃ : Esfuerzo del diseño en la tercera iteración.

|ED₄ : Esfuerzo del diseño en la cuarta iteración.

Capítulo 2: Propuesta de Solución

❖ Estimación de tiempo:

$$T = TD + TI + R$$

$$R = 10\%(TD + TI)$$

$$TD = \frac{ED}{TP}$$

$$TI = PE + \frac{1}{2}TD_1 + \frac{1}{2}TD_2 + \frac{1}{3}TD_3 + \frac{1}{3}TD_4$$

T: Tiempo de estimación de las pruebas.

TD: Tiempo de diseño de las pruebas.

TI: Tiempo de implementación de las pruebas.

R: Riesgo.

TP: Total de probadores.

TD₁ : Tiempo del diseño en la primera iteración.

TD₂ : Tiempo del diseño en la segunda iteración.

TD₃ : Tiempo del diseño en la tercera iteración.

TD₄ : Tiempo del diseño en la cuarta iteración.

- **Estimación de tiempo y esfuerzo para productos que requieran revisión del diseño e implementación de las pruebas de liberación.**

Como resultados obtenidos del modelo “Tormenta de Ideas” se tiene:

- ✓ Se estimará el tiempo y esfuerzo del diseño de los casos de prueba con los probadores que realizarán las revisiones.
- ✓ Cuando el producto que va a liberar, ya trae el diseño de los casos de prueba, se le realizará una revisión, para luego implementar dichos casos de prueba, esta revisión será 1/3 de la estimación

Capítulo 2: Propuesta de Solución

del diseño, teniendo en consideración que los probadores se demoran más diseñando los casos de prueba, que revisando.

❖ **Estimación de esfuerzo:**

$$E = ER + EI$$

$$ER = \frac{1}{3}ED$$

ER: Esfuerzo de la revisión del diseño.

❖ **Estimación de tiempo:**

$$T = TR + TI + R$$

$$TR = \frac{1}{3}TD$$

TR: Tiempo de la revisión del diseño.

2.3 Conclusiones Parciales

Se elaboró un procedimiento como propuesta, para que guíe la correcta estimación de tiempo y esfuerzo en las pruebas de liberación en el laboratorio de Calidad Central, para esta actividad se especificó la forma en que se debe desarrollar y aplicar el procedimiento, así como sus objetivos, fórmulas y restricciones, para facilitar la calidad del producto software en la UCI.

Capítulo 3: Validación del Procedimiento

Capítulo III: Validación del Procedimiento.

3.1 Introducción.

Teniendo en cuenta la importancia de la propuesta que se realiza en el presente trabajo, se utilizará este capítulo para llevar a cabo la validación de la investigación, mostrando estadísticamente la factibilidad del procedimiento presentado en el capítulo anterior. Para tal efecto se aplicó una técnica de investigación que utiliza un grupo de expertos para el diagnóstico, creación de posibles escenarios futuros, o búsqueda de soluciones a problemas planteados. El método se ajusta a las aplicaciones de toma de decisiones, pero debe estar adaptado en función del objetivo del estudio. El Método de Validación por Expertos (**Método Delphi**), es una técnica subjetiva basada en la opinión de expertos, para sondear la evolución del entorno de la propuesta.

3.2 Métodos de expertos.

Dentro de los métodos generales de prospectiva cabe destacar aquellos que se basan en la consulta a expertos, que reciben la denominación de métodos de expertos y utilizan como fuente de información un grupo de personas a las que se supone un conocimiento elevado de la materia que se va a tratar. (Croce 2006) (Astiragarra, 2000)

Estos métodos se emplean cuando se da alguna de las siguientes condiciones:

- No existen datos históricos con los que trabajar. Un caso típico de esta situación es la previsión de implantación de nuevas tecnologías.
- El impacto de los factores externos tiene más influencia en la evolución que el de los internos. Así, la aparición de una legislación favorable y reguladora y el apoyo por parte de algunas empresas a determinadas tecnologías, pueden provocar un gran desarrollo de éstas que de otra manera hubiese sido más lento.
- Las consideraciones éticas o morales dominan sobre las económicas y tecnológicas en un proceso evolutivo. En este caso, una tecnología puede ver dificultado su desarrollo si éste provoca un alto rechazo en la sociedad.
- Cuando el problema no se presta para el uso de una técnica analítica precisa.
- Cuando se desea mantener la heterogeneidad de los participantes a fin de asegurar la validez de los resultados.

Capítulo 3: Validación del Procedimiento

- Cuando el tema en estudio requiere de la participación de individuos expertos en distintas áreas del conocimiento.

3.3 Método Delphi.

El nombre de este método de prospección proviene del oráculo de Delphos, que se encontraba en la antigua Grecia, al que se acudía para hacer preguntas al Dios a través de una sacerdotisa. A pesar del carácter siempre ambiguo de las respuestas, el oráculo de Delphos era el que tenía mejor reputación por la certeza de sus predicciones. (Astiragarra, 2000)

Esa misma excelencia de prospección es la que pretende el Método Delphi. La consulta, a través de cuestionarios, para buscar un consenso en un grupo de expertos del ámbito se presupone una forma óptima y contrastada de evaluación. El método tiene cuatro etapas sucesivas de envíos de cuestionarios, de los que se sintetizan las respuestas para elaborar la siguiente consulta, hasta llegar a un resultado de consenso.

Tras un primer test de aproximación, en la segunda consulta los expertos deben volver a responder viendo los resultados de la primera y justificar sus divergencias con el grupo. En la tercera consulta, se pide a cada experto comentar los argumentos que disienten de la mayoría y en el cuarto turno se reclama la opinión sobre el consenso final. Se busca un consenso que sólo se verá truncado si las posturas son muy distantes, aunque también la presión por el acuerdo puede evitar una buena previsión. (Astiragarra, 2000)

Este método presenta tres características fundamentales, que le permiten garantizar la calidad de los resultados, para lanzar y analizar la Delphi:

1. **Anonimato:** Durante la puesta en práctica del método de experto, ningún experto conoce la identidad de los otros que componen el grupo de debate. Esto tiene una serie de aspectos positivos, como son:
 - Impide la posibilidad de que un miembro del grupo sea influenciado por la reputación de otro de los miembros o por el peso que supone oponerse a la mayoría. La única influencia posible es la de la congruencia de los argumentos.
 - Permite que un miembro pueda cambiar sus opiniones sin que eso suponga una pérdida de imagen.

Capítulo 3: Validación del Procedimiento

- El experto puede defender sus argumentos con la tranquilidad que da saber que en caso de que sean erróneos, su equivocación no va a ser conocida por los otros expertos.
- 2. Iteración y realimentación controlada:** La iteración se consigue al presentar varias veces el mismo cuestionario. Como, además, se van presentando los resultados obtenidos con los cuestionarios anteriores, se consigue que los expertos vayan conociendo los distintos puntos de vista y puedan ir modificando su opinión si los argumentos presentados les parecen más apropiados que los suyos.
- 3. Respuesta del grupo en forma estadística:** La información que se presenta a los expertos no es sólo el punto de vista de la mayoría, sino que se presentan todas las opiniones indicando el grado de acuerdo que se ha obtenido.

En la realización de un Delphi aparece una terminología específica:

Circulación: Es cada uno de los sucesivos cuestionarios que se presenta al grupo de expertos.

Cuestionario: El cuestionario es el documento que se envía a los expertos. No es sólo un documento que contiene una lista de preguntas, sino que es el documento con el que se consigue que los expertos interactúen, ya que en él se presentarán los resultados de anteriores circulaciones.

Panel: Es el conjunto de expertos que toma parte en el Delphi.

Moderador: Es la persona responsable de recoger las respuestas del panel y preparar los cuestionarios.

Fases: Antes de iniciar un Delphi se realizan una serie de tareas previas, como son:

- Delimitar el contexto y el horizonte temporal en el que se desea realizar la previsión sobre el tema en estudio. (Astigarraga 2000).
- Seleccionar el panel de expertos y conseguir su compromiso de colaboración. Las personas que sean elegidas no sólo deben ser grandes conocedores del tema sobre el que se realiza el estudio, sino que deben presentar una pluralidad en sus planteamientos. Esta pluralidad debe evitar la aparición de sesgos en la información disponible en el panel. (Astiragarra, 2000).
- Explicar a los expertos en qué consiste el método. Con esto se pretende conseguir la obtención de previsiones fiables, pues los expertos van a conocer en todo momento cuál es el objetivo de cada uno de los procesos que requiere la metodología. (Astiragarra, 2000).

Capítulo 3: Validación del Procedimiento

- En un Delphi clásico se pueden distinguir cuatro circulaciones, fases o etapas sucesivas de envíos de cuestionarios, de los que se sintetizan las respuestas para elaborar la siguiente consulta, hasta llegar a un resultado de consenso.

Esta técnica tiene la ventaja de eliminar el efecto líder de otros métodos de expertos, pues los encuestados son anónimos entre sí, pero es muy importante para un correcto resultado escoger bien a los testados y definir bien el campo de investigación, con preguntas precisas, cuantificables e independientes.

3.4 Validación del procedimiento.

Una de las ventajas del Delphi es la cuasi-certeza de obtener un consenso en el desarrollo de los cuestionarios sucesivos. La información recogida en el curso de la consulta acerca de acontecimientos, tendencias, rupturas determinantes en la evolución futura del problema estudiado, es generalmente rica y abundante.

La tramitación a través de encuestas en varias iteraciones es además discutible, puesto que solo los expertos que se salen de la norma deben justificar su posición. Sin embargo, se puede considerar también que la opinión de los divergentes es, en términos de prospectiva, más interesante que aquella de los que entran en el rango.

Para la aplicación práctica del método es necesario considerar:

- La selección del grupo de expertos.
- La elaboración del cuestionario.
- La evaluación de los resultados del proceso de validación.

3.4.1 Proceso de selección del grupo de expertos.

Una Delphi, como método de estructuración de un proceso de comunicación grupal efectivo a la hora de permitir a un grupo de individuos, como un todo, tratar un problema complejo, consiste en la selección de un grupo de expertos a los que se les pregunta su opinión sobre cuestiones referidas a acontecimientos del futuro. Las estimaciones de los expertos se realizan en sucesivas rondas, anónimas, al objeto de tratar de conseguir consenso, pero con la máxima autonomía por parte de los participantes.

Capítulo 3: Validación del Procedimiento

Un experto es una persona o un grupo de personas que tiene un gran conocimiento sobre el tema tratado, que puede emitir un criterio concluyente de cualquier problema y emitir valoraciones importantes con un alto nivel de conocimiento. Para la selección del grupo de expertos es necesario determinar las áreas del conocimiento que deben dominar, en este caso, partiendo del problema planteado, se determinó que los expertos debían dominar:

- Proceso de Desarrollo del Software.
- Proceso Unificado del Software.
- Flujo de diseño e implementación de pruebas.

A continuación se confeccionó un listado inicial de expertos teniendo en cuenta su experiencia en las áreas identificadas. El listado estuvo conformado por 12 expertos: 2 directivos, 3 especialistas de Calidad Central que han realizado estudios sobre el tema, 2 especialistas en métricas, 1 máster, 2 jefes de departamentos centrales y 2 asesores de calidad. Luego de la solicitud del consentimiento de los posibles involucrados para participar en la validación del procedimiento, 8 estuvieron de acuerdo.

De manera particular, las características de los expertos, influyen en los resultados obtenidos y dan la medida del grado de confiabilidad del mismo. La calificación técnica, la capacidad de emitir una decisión, la tenencia de conocimientos específicos sobre el tema a evaluar y la disposición a participar como parte del grupo de expertos, son temas que deben tenerse en cuenta antes de iniciar el proceso de validación.

Todas las personas seleccionadas para participar en el Panel de Expertos cuentan con los conocimientos necesarios para emitir una valoración y están o han estado vinculados de forma directa al proceso de pruebas de liberación en la UCI. Los conocimientos de los expertos encuestados avalan la participación de todos en la evaluación de la propuesta.

A continuación se muestra el dominio de los expertos en las áreas del conocimiento requeridas para formar parte del Panel de Expertos:

Capítulo 3: Validación del Procedimiento

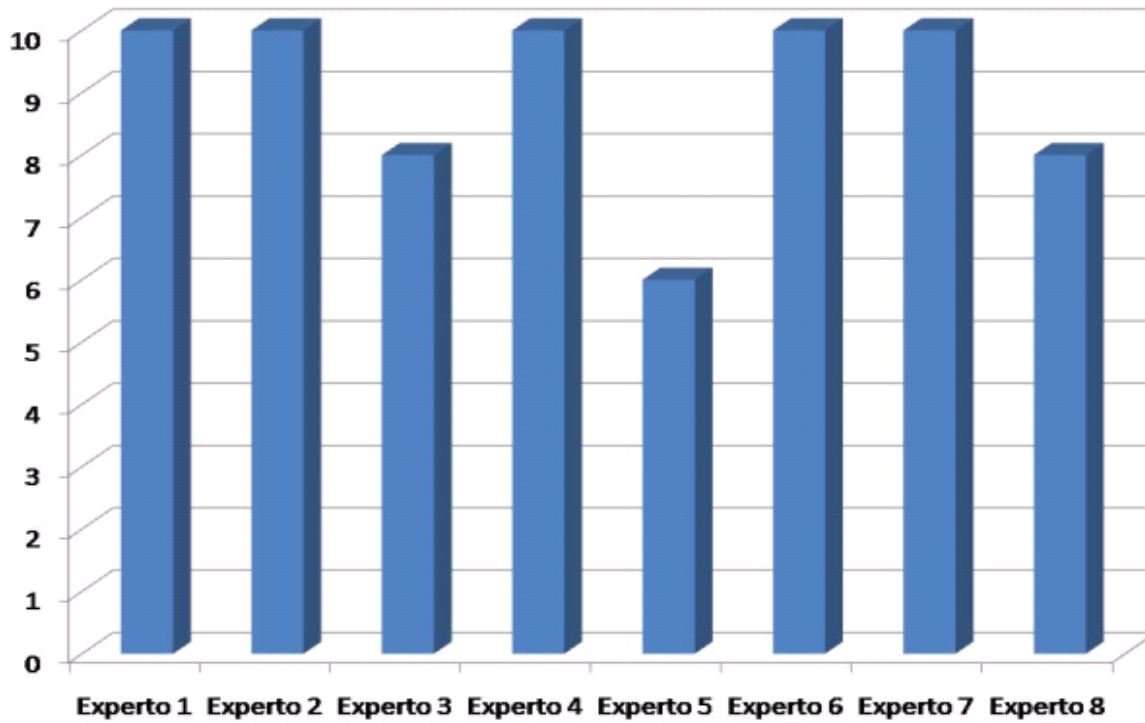


Gráfico 2: Grado de conocimiento de los expertos sobre el Proceso de Desarrollo del Software.

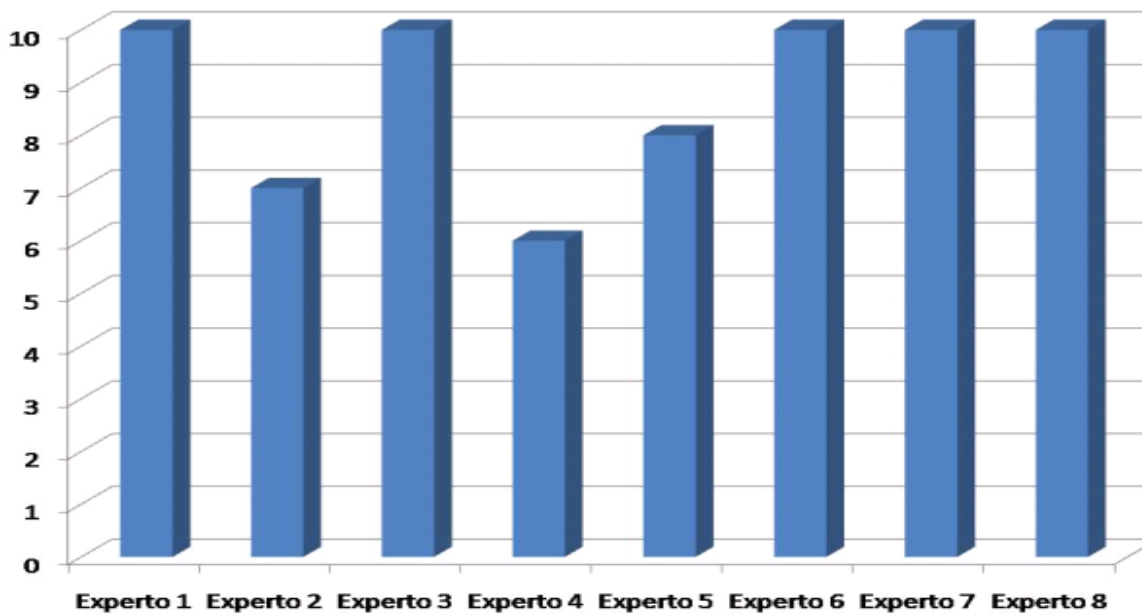


Gráfico 3: Grado de conocimiento de los expertos sobre el Proceso Unificado del Software.

Capítulo 3: Validación del Procedimiento

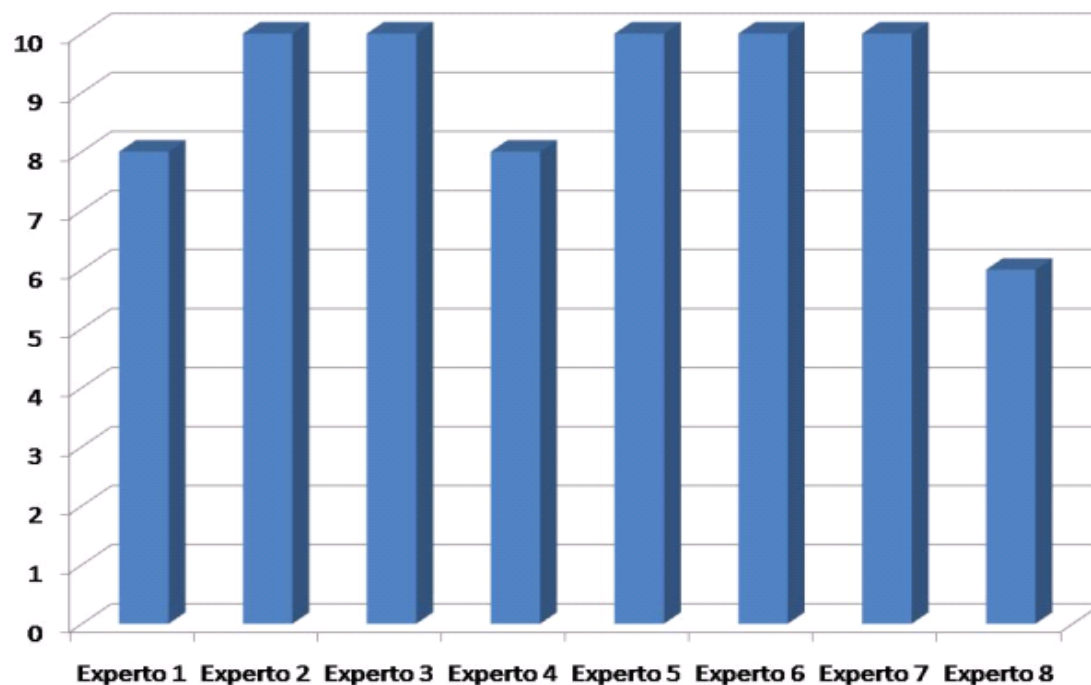


Gráfico 4: Grado de conocimiento de los expertos sobre el flujo de diseño e implementación de pruebas.

3.4.2 Elaboración del Cuestionario.

Las preguntas se refieren a las probabilidades que existen de una estimación más real del tiempo y esfuerzo de las pruebas de liberación con la aplicación de este procedimiento. Esto depende de la elaboración del cuestionario y de la elección de los expertos consultados.

Para su confección se tuvo en cuenta realizar las preguntas con un enfoque investigativo, centrándose en los objetivos fundamentales que persigue el procedimiento propuesto y permitiendo a los expertos expresar su criterio personal del tema. **Ver Anexo 5.**

En este proceso, el procedimiento fue presentado a los expertos por escrito y de forma personal, se completaron las respuestas y se realizaron las preguntas pertinentes, según lo analizado por cada uno de ellos durante el estudio de la propuesta.

Capítulo 3: Validación del Procedimiento

3.4.3 Resultados de la validación.

Con el fin de validar el procedimiento para estimar tiempo y esfuerzo de las pruebas de liberación en los proyectos de la UCI se aplicó el cuestionario mostrado en el **Anexo 5**, sus objetivos fueron:

- Determinar si la propuesta realizada es la correcta para darle solución al problema planteado con anterioridad.
- Comprobar la efectividad, completitud y firmeza del procedimiento.
- Identificar aspectos desacertados y recomendaciones que permitan mejorar la propuesta.

➤ **Resultados del cuestionario.**

En la primera pregunta del cuestionario se evaluaron los elementos (complejidad de los casos de uso y la experiencia de los probadores) que se tuvieron en cuenta en el procedimiento de estimación de tiempo y esfuerzo de las pruebas de liberación, arrojando como resultado:

- ✓ Cinco expertos evaluaron como necesarios y suficiente los elementos tenidos en cuenta en el procedimiento.
- ✓ Tres expertos evaluaron como necesarios pero no suficientes los elementos tenidos en cuenta en el procedimiento.
- ✓ Ningún experto consideró innecesarios los elementos tenidos en cuenta en el procedimiento.

Por tanto, se consideró que son válidos los elementos tenidos en cuenta en el procedimiento de estimación de tiempo y esfuerzo de las pruebas de liberación.

Capítulo 3: Validación del Procedimiento

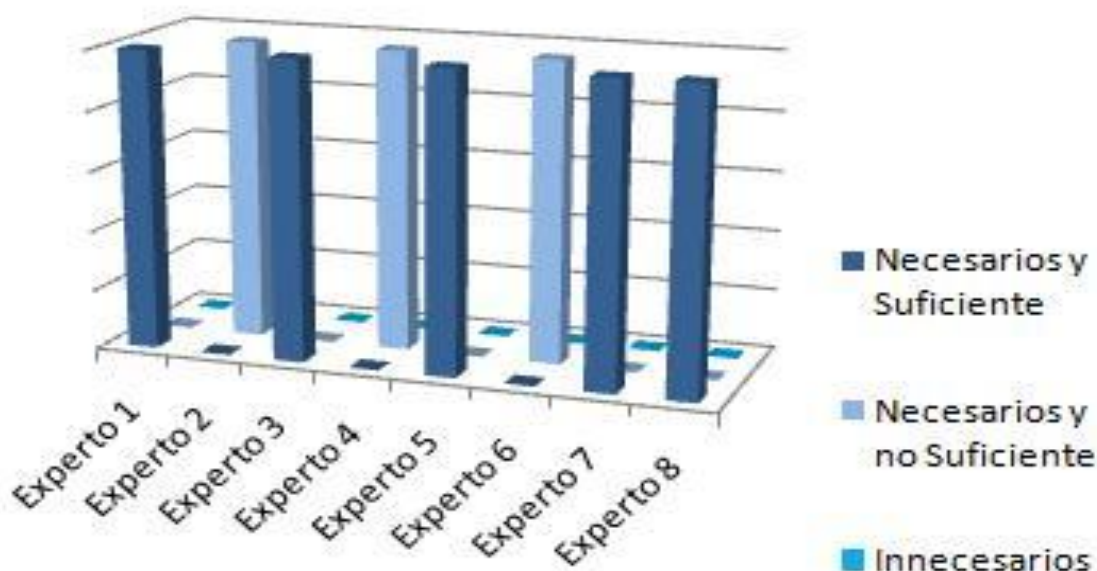


Gráfico 5: Elementos tenidos en cuenta en el procedimiento.

En la segunda pregunta del cuestionario se evaluaron los elementos (cantidad de flujos y elementos de E/S) que se tuvieron en cuenta para clasificar la complejidad de los casos de uso, arrojando como resultado:

- ✓ Seis expertos evaluaron como necesarios y suficiente, los elementos tenidos en cuenta para clasificar la complejidad de los casos de uso.
- ✓ Dos expertos evaluaron como necesarios pero no suficientes, los elementos tenidos en cuenta para clasificar la complejidad de los casos de uso.
- ✓ Ningún experto consideró innecesarios los elementos tenidos en cuenta para clasificar la complejidad de los casos de uso.

Por tanto, se consideró que son válidos los elementos tenidos en cuenta para clasificar la complejidad de los casos de uso.

Capítulo 3: Validación del Procedimiento

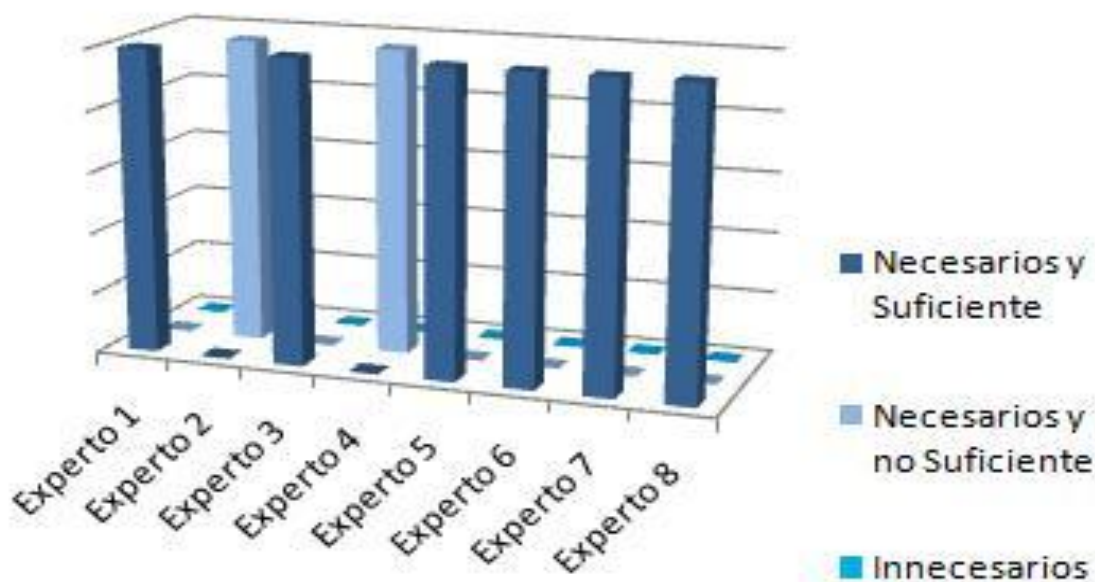


Gráfico 6: Elementos tenidos en cuenta para clasificar la complejidad de los casos de uso.

En la tercera pregunta del cuestionario se evaluaron los elementos (conocimientos y habilidades) que se tuvieron en cuenta para clasificar la experiencia de los probadores, arrojando como resultado:

- ✓ Seis expertos evaluaron como necesarios y suficiente, los elementos tenidos en cuenta para clasificar la experiencia de los probadores.
- ✓ Dos expertos evaluaron como necesarios pero no suficientes, los elementos tenidos en cuenta para clasificar la experiencia de los probadores.
- ✓ Ningún experto consideró innecesarios los elementos tenidos en cuenta para clasificar la experiencia de los probadores.

Por tanto, se consideró que son válidos los elementos tenidos en cuenta para la clasificación de la experiencia de los probadores.

Capítulo 3: Validación del Procedimiento

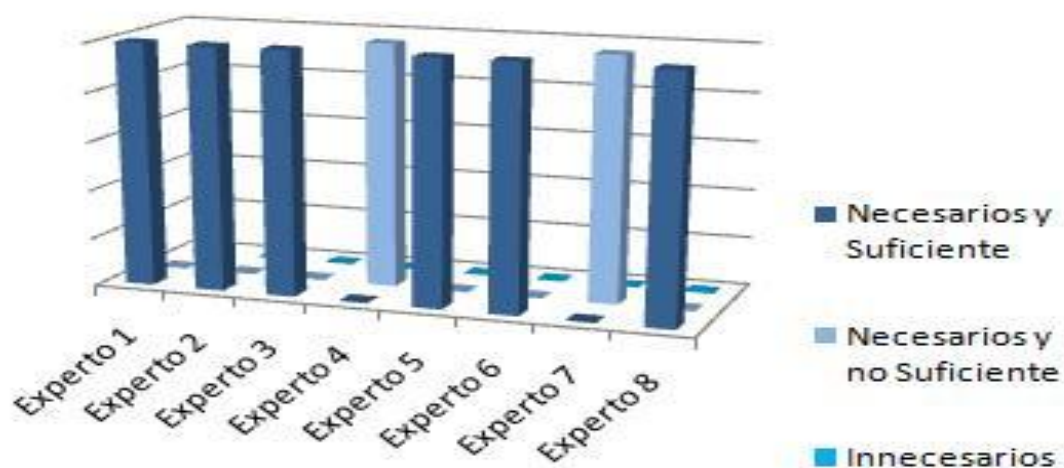


Gráfico 7: Elementos tenidos en cuenta para clasificar la experiencia de los probadores.

En la cuarta pregunta del cuestionario se muestra el ajuste del procedimiento con el proceso de pruebas de liberación que se aplica en los proyectos de la universidad, arrojando como resultado:

- ✓ Seis expertos determinaron que el procedimiento se ajusta totalmente al proceso de pruebas de liberación que se aplica en los proyectos de la universidad.
- ✓ Dos expertos determinaron que el procedimiento se ajusta en gran medida al proceso de pruebas de liberación que se aplica en los proyectos de la universidad.
- ✓ Ningún experto determinó que el procedimiento se ajusta poco al proceso de pruebas de liberación que se aplica en los proyectos de la universidad.

Por tanto, se consideró que el procedimiento se ajusta al proceso de pruebas de liberación que se aplica en los proyectos de la universidad.

Capítulo 3: Validación del Procedimiento

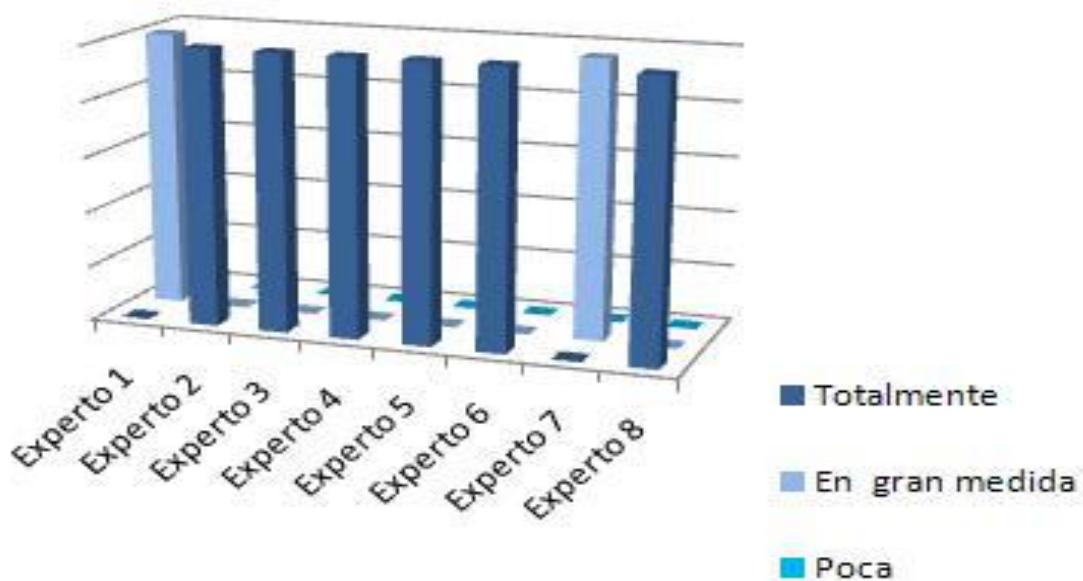


Gráfico 8: Ajuste del procedimiento al proceso de pruebas de liberación.

En la quinta pregunta del cuestionario se propone el procedimiento como una mejora a la actual forma de estimar el tiempo y esfuerzo de las pruebas de liberación, arrojando como resultado:

- ✓ Cinco expertos evaluaron el procedimiento propuesto como una mejora total, a la actual forma de estimar tiempo y esfuerzo de las pruebas de liberación.
- ✓ Tres expertos consideraron que el procedimiento propuesto mejorará en gran medida la actual forma de estimar tiempo y esfuerzo de las pruebas de liberación.
- ✓ Ningún experto consideró que el procedimiento propuesto traerá poca mejora a la actual forma de estimar tiempo y esfuerzo de las pruebas de liberación.

Por tanto, se consideró que es válido el procedimiento como mejora a la actual forma de estimación de tiempo y esfuerzo de las pruebas de liberación.

Capítulo 3: Validación del Procedimiento

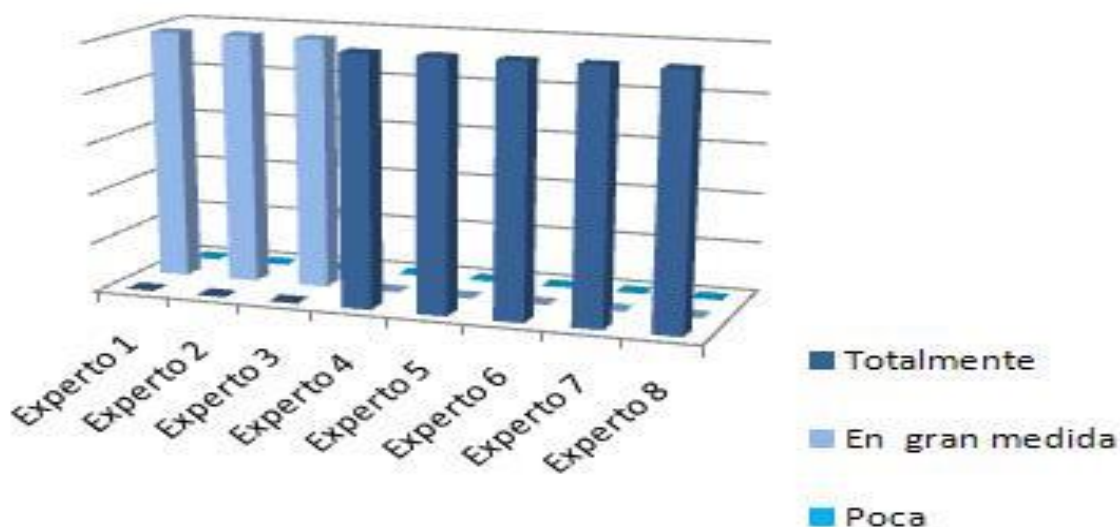


Gráfico 9: Procedimiento como mejora a la actual forma de estimación de tiempo y esfuerzo de las pruebas de liberación.

En la sexta pregunta del cuestionario se propone la efectividad de la aplicación del procedimiento, en la planificación de las pruebas de liberación de un proyecto, arrojando como resultado:

- ✓ Seis expertos consideraron que la aplicación del procedimiento propuesto, tendrá una efectividad total en la planificación de las pruebas de liberación de un proyecto.
- ✓ Dos expertos consideraron que la aplicación del procedimiento propuesto, será efectivo en gran medida en la planificación de las pruebas de liberación de un proyecto.
- ✓ Ningún experto consideró que la aplicación del procedimiento propuesto, no tendrá efectividad en la planificación de las pruebas de liberación de un proyecto.

Por tanto, se consideró que la aplicación del procedimiento propuesto, será efectivo en la planificación de las pruebas de liberación de un proyecto.

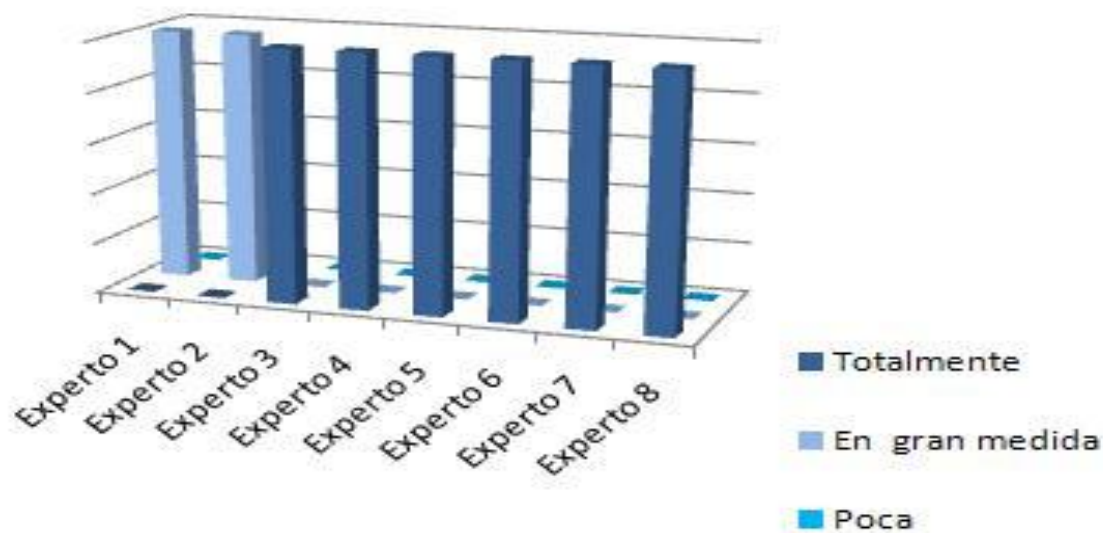


Gráfico 10: Efectividad de la aplicación del procedimiento, en la planificación de las pruebas de liberación de un proyecto.

Corrección y completitud del procedimiento.

Para valorar la corrección y completitud del procedimiento se elaboraron las preguntas 1, 2 y 3 del cuestionario, cada una de ellas con importantes muestras que evalúan los elementos representativos para estimar tiempo y esfuerzo de las pruebas de liberación. En estos puntos se pone a consideración de los expertos la utilidad de dichos elementos como factor importante en la propuesta realizada. El resultado se muestra en el **Gráfico 12**.

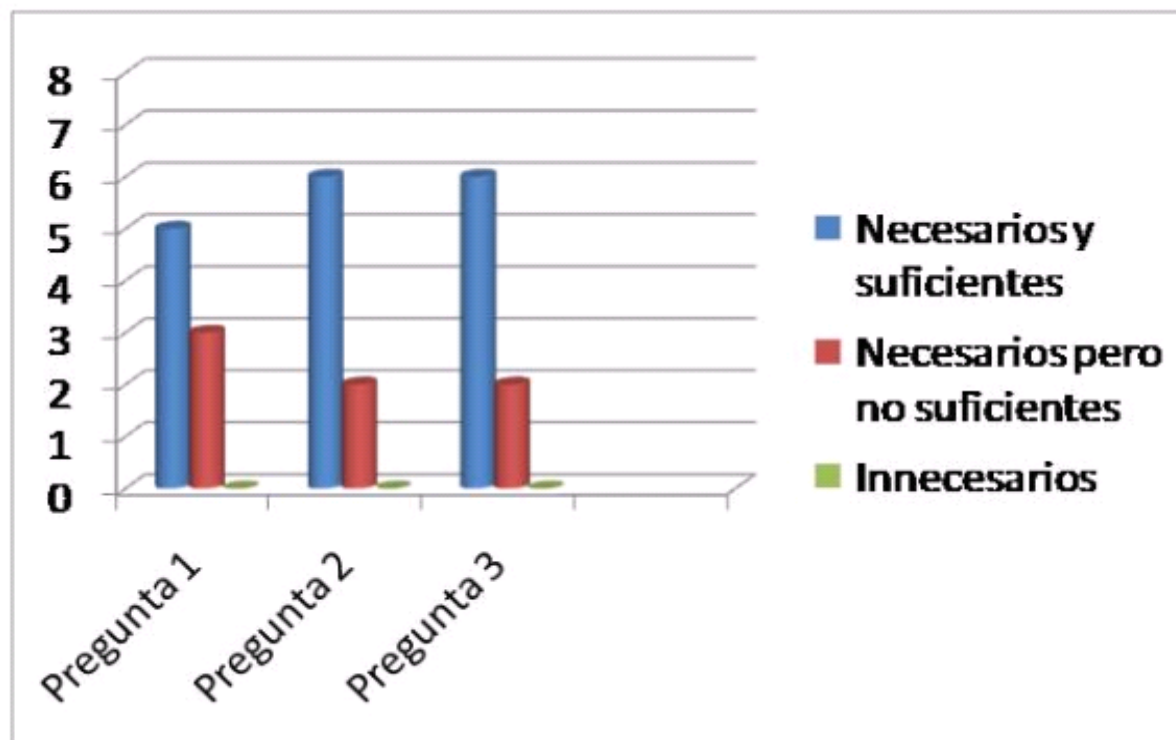


Gráfico 11: Elementos que se tuvieron en cuenta para estimar duración y esfuerzo de las pruebas de liberación.

Utilidad del procedimiento.

Las preguntas 4, 5 y 6 están referidas a la aplicación del procedimiento propuesto a proyectos del laboratorio de Calidad Central, y si el mismo mejora la actual forma de realizar la estimación de tiempo y esfuerzo de las pruebas de liberación. Las respuestas a los puntos podían ser categorizadas (Totalmente/En gran medida/Poca). Los resultados se muestran en el **Gráfico 13**.

Capítulo 3: Validación del Procedimiento

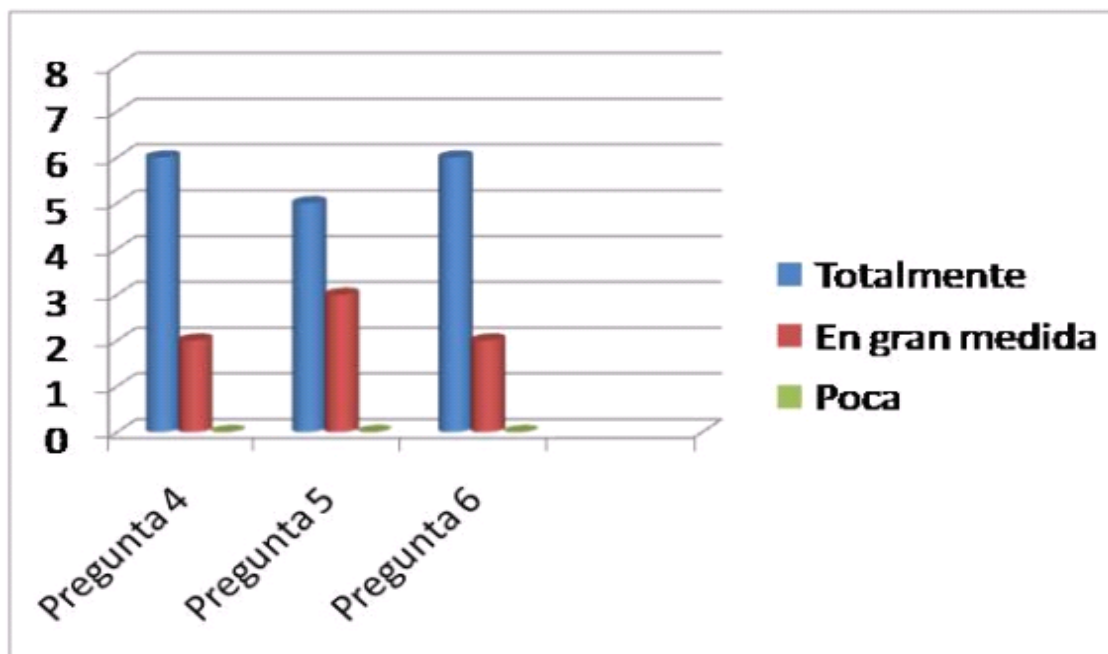


Gráfico 12: Procedimiento como mejora.

En las preguntas 7 y 8 se recogen otros criterios emitidos por los expertos que ayudan a la mejora del procedimiento propuesto. A continuación se exponen las siguientes opiniones:

- Sería una buena experiencia la puesta en práctica del procedimiento en un proyecto real.
- Enseña una nueva manera de estimar el tiempo y esfuerzo que puede ser útil para los equipos de proyecto que decidan aplicarlas.
- El procedimiento es novedoso, constituye una propuesta interesante para realizar estimación de tiempo y esfuerzo en las pruebas de liberación.
- La utilización de este procedimiento puede viabilizar la estimación de tiempo y esfuerzo de las pruebas de liberación.
- Representa un adelanto en el tema de estimación de tiempo y esfuerzo en las pruebas de liberación de software.
- La aplicación de este procedimiento supone un avance en la planificación de las pruebas de liberación, pues ya se tiene una estimación inicial, que puede ser cercana a la realidad en

Capítulo 3: Validación del Procedimiento

dependencia de la seriedad con que se clasifiquen los probadores y los casos de uso asignados al proceso.

El grado de concordancia entre los expertos se determinó a través del coeficiente de Kendall, una de las técnicas no paramétricas para medir el grado de correlación y asociación entre varios conjuntos (k) de N entidades donde:

K = 8: Número de especialistas encuestados.

N = 8: Cantidad de preguntas con la que cuenta la encuesta.

Las hipótesis serían:

H0: No hay concordancia entre los expertos encuestados.

H1: Hay concordancia entre los expertos encuestados.

Como (N > 7) se utiliza la fórmula siguiente:

$$\chi^2 = k(N - 1) * W$$

Con N-1 grados de libertad.

El Chi cuadrado calculado es igual a 21.000 y la región crítica sería $\chi^2 > \chi^2(k, n-1)$.

En la tabla correspondiente, la región de rechazo sería: $\chi^2 > 18.5$

Como el estadígrafo calculado (21.0) es mayor que el tabulado (18.5) rechazamos la hipótesis nula, ya que la W es significativamente distinta de cero. Ver Figura 1

N	8
Kendall's W ^a	,375
Chi-Square	21,000
df	7
Asymp. Sig.	,004

a. Kendall's Coefficient of Concordance

Figura 1. Resultado de la aplicación del Coeficiente de Concordancia de Kendall.

Capítulo 3: Validación del Procedimiento

3.5 Conclusiones parciales.

Delphi es sin duda una técnica que desde hace unos cuarenta años ha sido objeto de múltiples aplicaciones en el mundo entero. A partir del procedimiento original, se han desarrollado otras anteriores. Permite afirmar de manera general que el procedimiento fue evaluado por los expertos como: válido, correcto y completo, para desarrollar la estimación de tiempo y esfuerzo de las pruebas de liberación en los proyectos productivos de la universidad en el laboratorio de Calidad Central.

Conclusiones

Conclusiones

En el presente trabajo se ha descrito e ilustrado un procedimiento para la estimación de tiempo y esfuerzo de las pruebas de liberación en el laboratorio de Calidad Central. Este procedimiento puede ser aplicado por el personal del laboratorio para tener una aproximación más cercana del tiempo que demorarán las pruebas que se le realizarán al producto que se va a liberar. El procedimiento tiene características relevantes integrando aspectos que son fundamentales para estimar. Es bastante simple y se encuentra bien estructurado.

En el desarrollo de este trabajo se cumplieron todos los objetivos planteados:

- Se realizó una investigación sobre técnicas usadas en el mundo para estimar el tiempo y esfuerzo del desarrollo del software, las cuales no se ajustan al proceso de pruebas de liberación de la UCI.
- Se propuso un procedimiento para estimar tiempo y esfuerzo de las pruebas de liberación en el laboratorio de Calidad Central.
- El procedimiento planteado fue evaluado y aceptado, usando como herramienta Criterio de Expertos, específicamente Método Delphi.
- Se demostró el valor significativo que representa la adecuada estimación de tiempo y esfuerzo de las pruebas de liberación.
- El procedimiento propuesto contribuye a elevar la calidad de la estimación de las pruebas de liberación en el laboratorio de la Dirección de Calidad, brindando elementos a los especialistas de prueba, para estimar objetivamente el diseño e implementación de las pruebas.
- Se demostró el valor significativo que representa la adecuada estimación de tiempo y esfuerzo de las pruebas de liberación.
- El procedimiento propuesto contribuye a elevar la calidad de la estimación de las pruebas de liberación en el laboratorio de la Dirección de Calidad, brindando elementos a los especialistas de prueba, para estimar objetivamente el diseño e implementación de las pruebas.

Recomendaciones

Recomendaciones:

- Desarrollar un sistema que permita automatizar todo o parte del procedimiento estudiado.
- Trabajar en cómo realizar una buena planificación de las sesiones de trabajo, teniendo en cuenta la experiencia de los probadores y la complejidad de los casos de uso.
- Aplicar el procedimiento a un proyecto, con el objetivo de estimar el tiempo y esfuerzo de las pruebas.

Bibliografía

Bibliografía

- Grimán Anna.C, Pérez María, Mendoza Luis E. 2009.** Estrategia de Pruebas para Software OO que garantiza Requerimientos No Funcionales. Caracas venezuela : s.n., 2009.
- Astiragarra. 2000.** 2000.
- Bartle, Phil. 2009.** Tormenta de Ideas:Procedimiento y Procesos. [trad.] Lourdes sada. 29 de 4 de 2009. Tormenta de Ideas.
- Caldas, Universidad Distrital Francisco José de. 2006.** VISION GENERAL DE LAS PRUEBAS DE SOFTWARE. 2006.
- Calisoft. 2008.** Centro para la Excelencia en el desarrollo de proyectos tecnológicos. La Habana, Cuba : s.n., 2008.
- Croce. 2006.** 2006.
- Davis, A. 1995.** Principles of Software Development. 1995.
- Domingo J, Delgado M. 2000.** Modelo de Gestión por Competencia. 2000.
- Edward. 1986.** 1986.
- Entorno Virtual de Aprendizaje, 2006.** Tercer año.Ingeniería de Software.Conferencia 5. [En línea] 2006. [Citado el: 5 de 12 de 2008.] <http://teleformación.uci.cu>.
- Estevez, María de Lourdez, Arrieta José Joaquín Gallastequi. 2006.** El Método Delphi su implementación en una estrategia didáctica para la enseñanza de las demostraciones geométricas. 2006.
- Franco, JA. 2002.** Arquitectura de Software. 2002.
- González, C. 2003.** Conceptos generales de calidad total. [En línea] 2003. <http://www.ilustrados.com/pulicaciones>.
- Halten, KJ. 1987.** Direcciones estratégicas. 1987.
- Hernández Rolando Alfredo León, Coello Sayda Gonzalez. 2002.** El paradigma cuantitativo de la investigación científica. Ciudad de La Habana : s.n., 2002.
- Clayton. 2005.** Herramientas de apoyo al proceso de Ingeniería de Software en pequeñas empresas. 2005, Mastermagazine, pág. 1.

Bibliografía

- Informatica, Facultad de. 2002.** Unidades temáticas de Ingeniería del Software. Estrategias de prueba del software. Segunda ed. 2002.
- Diaz JF. 2008.** Métodos de Prueba de Programas. 2008.
- Rojas Johanna, Barrios Emilio. 2007.** Métodos de Prueba de Caja Negra. 2007.
- John V, John M. 2008.** La Importancia de la Prueba del Software. 2008.
- Kynetia. 2007.** Tipos de Pruebas. 2007.
- Chaves, Michael Arias. 2007.** La Ingeniería de Requerimientos y su importancia en el desarrollo de proyectos de software. 10, Costa Rica : s.n., 26 de 07 de 2007, Revista InterSedes, Vol. V6.
- Lander JA, Del Moral Daniel. 1999.** Métodos de Prueba. 1999.
- Collado Manuel. marzo,2003.** Pruebas de software. Técnicas de prueba del software. Estrategias de prueba de software. marzo,2003.
- Métodos de prospectiva. Criterio de Expertos. [En línea] [Citado el: 15 de 03 de 2009.]
<http://www.gtictic.ssr.upm.es>.
- Myers, G. 1979.** The Art of Software Testing. 1979.
- Parrado, Arenas Alexandra. 2008.** Publicaciones X Semestre ING. Sistemas. 19 de 05 de 2008.
- Prado, Elena Raja. 2007.** Pruebas de Software. 2007. Vol. Vol 1.
- Pressman, R.S. 2005.** Un enfoque práctico. Quinta edición. s.l. : Félix Varela, 2005. Vol. 1.
- Blanco Romero Alberto, Izquierdo Cecilia Paula Moreno. 2004.** El Método Delphi. 2004, pág. 10.
- RUP. 2003.** Rational Software Corporation. 2003.
- Gonzalez, Pablo Javier Tuya. 2007.** Sobre el Taller de Pruebas en Ingeniería del Software. 003, Madrid, España : s.n., 2007, REICIS: Revista Española de Innovación, Calidad e Ingeniería del Software., págs. 42-43.
- Sommerville. 2006.** Ingeniería de Software. Séptima ed. 2006. Vol. 1.
- Stellman A. 2006.** Applied Software Project.
- Vilar, Roser. 2008.** Las pruebas de calidad de software, una asignatura pendiente. España : s.n., 09 de 2008.

Bibliografía

Wodzislawski Mónica, Pérez Beatriz, Pittier Amparo, Travieso Mariana. 2006. Prueba Exploratoria en la práctica. 2006.

Y, Roberto. 2007. Los 5 tipos de pruebas de software. 2007.

Iglesias Yohandy Cruz, Velazquez Alionuzca Cintra. 2008. Procedimiento para modelar el negocio a través de la gestión de procesos. La Habana, Cuba : s.n., 19 de 05 de 2008.

Anexos

Anexos

Anexo 1:

Entrevista realizada al Jefe del Laboratorio de Pruebas de Calidad Central.

Objetivos de la entrevista:

- Conocer el proceso de estimación de tiempo y esfuerzo de las pruebas de liberación en el laboratorio de la Dirección de Calidad.
- Conocer la utilidad del método de estimación utilizado actualmente.
- Conocer la necesidad que existe de un nuevo método o procedimiento que permita estimar lo más exacto posible el tiempo y esfuerzo de las pruebas de liberación.

Cuestionario:

1. ¿Se utilizan los métodos que existen actualmente para estimar el tiempo y esfuerzo de las pruebas de liberación?
2. ¿Qué métodos utilizan para estimar tiempo y esfuerzo de las pruebas de liberación en el laboratorio de Calidad Central?
3. ¿Corresponde dicha estimación con los resultados obtenidos en la realidad?
4. ¿Sería factible la utilización de un nuevo método o procedimiento que permita estimar de forma eficiente el tiempo y esfuerzo de las pruebas de liberación, en el laboratorio de Calidad Central?

Anexo 2:

Entrevista realizada al Jefe del laboratorio de Calidad Central.

Objetivos de la entrevista:

- Conocer la estructura de la Dirección de Calidad Central.
- Conocer el proceso de pruebas de liberación en la UCI.

Cuestionario:

1. ¿Cuándo y por qué surge el laboratorio de Calidad Central?
2. ¿Por qué surgen los grupos de calidad en las facultades?
3. ¿Cómo está conformada la Dirección de Calidad?

Anexos

4. ¿Cuál es la relación existente entre los grupos de calidad de las facultades y el laboratorio de Calidad Central?
5. ¿Cuál es el procedimiento que se lleva a cabo para la liberación de un proyecto?

Anexo 3:

Entrevista realizada a un grupo de especialistas del laboratorio de Calidad Central.

Objetivos de la entrevista:

- Conocer las clasificaciones de los casos de uso y de la experiencia de los probadores.
- Conocer los elementos que puedan determinar la clasificación de la experiencia de los probadores y los casos de uso.
- Conocer el tiempo que emplean en revisar los casos de prueba y el que emplean en las iteraciones de la implementación de las pruebas.
- Asociar un riesgo a la estimación.

Cuestionario:

1. ¿Cómo clasificaría usted los casos de uso?
2. ¿Qué elementos tendría en cuenta para su clasificación?
3. ¿Si estos elementos son cuantificables, especifique el rango?
4. ¿Qué peso usted le asignaría a la clasificación de los casos de uso?
5. ¿Cómo clasificaría usted a los probadores de acuerdo a su experiencia?
6. ¿Qué elementos tendría en cuenta en la gestión por competencia para su clasificación?
7. ¿Qué peso le asignaría a la experiencia de los probadores?
8. La estimación de tiempo y esfuerzo de la implementación consta de cuatro iteraciones. ¿Cuales de estas iteraciones requieren de mayor esfuerzo y tiempo para usted?
9. ¿Qué % le daría usted al riesgo asociado a la estimación de tiempo?
10. ¿Qué tiempo y esfuerzo emplean para las revisiones de los casos de prueba?

Anexos

Anexo 4:

Proyecto	Casos de Uso	Sesiones	Flujo Básico	Flujos Alternativos	Elementos de Entrada y/o Salida
Aduana	CU # 1	1	1	1	117
	CU # 2	1	1	1	33
	CU # 3	1	1	1	19
	CU # 4	1	1	1	20
	CU # 5	1	1	1	20
	CU # 6	1	1	1	18
	CU # 7	1	1	0	20
	CU # 8	1	1	1	19
	CU # 9	1	1	0	127
	CU # 10	1	1	0	120
	CU # 11	1	1	0	4
	CU # 12	1	1	0	1
	Cu # 13	1	1	0	126
	Cu # 14	1	1	0	0
CICPC	CU # 1	4	1	8	25
	CU # 2	2	1	8	46

Anexos

	CU # 3	1	1	12	32
	CU # 4	4	1	4	185
	CU # 5	1	1	10	49
	CU # 6	1	1	10	21
	CU # 7	5	1	14	82
	CU # 8	4	1	4	14
	CU # 9	1	1	11	4
	CU # 10	1	1	8	31
	CU # 11	1	1	6	63
	CU # 12	6	1	16	142
	CU # 13	2	1	8	10
	CU # 14	1	1	7	16
	CU # 15	4	1	6	30
	CU # 16	1	1	2	6
	CU # 17	2	1	6	12
	CU # 18	6	1	5	89
Identidad	CU # 1	1	1	1	6
	CU # 2	2	1	0	4
	CU # 3	3	1	0	6

Anexos

	CU # 4	3	1	0	9
	CU # 5	1	1	10	15
	CU # 6	1	1	14	30
	CU # 7	1	1	2	5
	CU # 8	1	1	1	5
	CU # 9	1	1	1	6
	CU # 10	1	1	2	7
	CU # 11	1	1	2	6
	CU # 12	7	1	2	6
	CU # 13	1	1	5	7
Prisiones	CU # 1	1	1	1	28
	CU # 2	1	1	1	15
	CU # 3	1	1	1	55
	CU # 4	1	1	1	25
	CU # 5	1	1	1	9
	CU # 6	1	1	1	5
	CU # 7	1	1	1	33
	CU # 8	1	1	1	22
	CU # 9	1	1	1	22

Anexos

	CU # 10	1	1	1	9
	CU # 11	1	1	1	16
	CU # 12	1	1	1	16
SIGESC	CU # 1	1	1	9	2
	CU # 2	1	1	1	2
	CU # 3	1	1	1	2
	CU # 4	1	1	3	2
	CU # 5	1	1	1	6
	CU # 6	1	1	2	18
	CU # 7	1	1	1	15
	CU # 8	1	1	0	7
	CU # 9	1	1	1	4
	CU # 10	1	1	3	13
	CU # 11	1	1	2	24
	CU # 12	1	1	2	3
	CU # 13	3	1	7	30
	CU # 14	1	1	4	19
	CU # 15	1	1	5	22
	CU # 16	2	1	2	2

Anexos

	CU # 17	1	1	2	1
	CU # 18	1	1	2	2
	CU # 19	1	1	1	24
	CU # 20	1	1	11	8
	CU # 21	1	1	2	2
	CU # 22	1	1	0	4
	CU # 23	1	1	1	3
	CU # 24	1	1	8	7
	CU # 25	1	1	5	60
	CU # 26	1	1	4	16
	CU # 27	1	1	5	18
	CU # 28	1	1	8	32
	CU # 29	1	1	2	13
	CU # 30	1	1	9	68
	CU # 31	1	1	8	53
	CU # 32	1	1	9	38
	CU # 33	1	1	5	8
	CU # 34	3	1	0	8
	CU # 35	1	1	0	20

Anexos

	CU # 36	1	1	1	6
	CU # 37	4	1	0	
	CU # 38	1	1	6	30
	CU # 39	1	1	2	6
	CU # 40	1	1	3	13
	CU # 41	1	1	4	2
	CU # 42	1	1	3	14
	CU # 43	1	1	1	22
	CU # 44	1	1	1	10
	CU # 45	1	1	1	6
	CU # 46	1	1	4	20
	CU # 47	1	1	3	3
	CU # 48	1	1	4	4
	CU # 49	1	1	4	80
	CU # 50	1	1	6	51
	CU # 51	1	1	2	9
	CU # 52	1	1	1	24
	CU # 53	1	1	1	1
	CU # 54	1	1	4	1

Anexos

	CU # 55	1	1	1	9
	CU # 56	1	1	2	7
	CU # 57	6	1	0	15
	CU # 58	1	1	1	20
	CU # 59	1	1	1	5
	CU # 60	1	1	1	8
	CU # 61	1	1	4	10
	CU # 62	1	1	1	10
	CU # 63	1	1	1	8
	CU # 64	1	1	7	12
	CU # 65	1	1	5	20
	CU # 66	1	1	6	23
	CU # 67	1	1	6	30
	CU # 68	1	1	3	6
	CU # 69	1	1	2	15
	CU # 70	2	1	0	4
	CU # 71	1	1	1	5
	CU # 72	1	1	3	6
	CU # 73	1	1	3	13

Anexos

	CU # 74	1	1	1	2
	CU # 75	1	1	3	2
	CU # 76	1	1	8	13
	CU # 77	1	1	5	25
	CU # 78	1	1	1	4
	CU # 79	1	1	1	12
	CU # 80	1	1	5	3
	CU # 81	1	1	4	3
	CU # 82	2	1	1	4
	CU # 83	1	1	1	8
	CU # 84	1	1	3	14
	CU # 85	1	1	2	7
	CU # 86	1	1	4	18
	CU # 87	1	1	1	8
	CU # 88	1	1	4	16
	CU # 89	1	1	1	22
	CU # 90	1	1	4	13
	CU # 91	1	1	6	21
	CU # 92	1	1	3	2

Anexos

	CU # 93	4	1	0	13
	CU # 94	1	1	3	18
	CU # 95	1	1	2	2
	CU # 96	1	1	2	18
	CU # 97	1	1	1	4
	CU # 98	1	1	3	13
	CU # 99	1	1	3	16
	CU # 100	1	1	2	8
	CU # 101	1	1	1	5
	CU # 102	2	1	3	2
	CU # 103	1	1	2	8
	CU # 104	1	1	3	7
	CU # 105	1	1	3	4
	CU # 106	1	1	2	8
	CU # 107	1	1	1	3
	CU # 108	1	1	1	18
	CU # 109	1	1	2	3
	CU # 110	1	1	2	3
	CU # 111	1	1	2	15

Anexos

	CU # 112	1	1	0	18
	CU # 113	1	1	0	20
	CU # 114	1	1	2	15
	CU # 115	1	1	2	7
	CU # 116	1	1	4	25
	CU # 117	4	1	1	5
	CU # 118	1	1	3	6
	CU # 119	1	1	5	8
	CU # 120	1	1	4	10
	CU # 121	1	1	2	9
	CU # 122	1	1	2	6
	CU # 123	1	1	2	12
	CU # 124	1	1	19	18
	CU # 125	1	1	1	3
	CU # 126	1	1	3	8
	CU # 127	1	1	4	12
	CU # 128	1	1	1	12
	CU # 129	1	1	5	8
	CU # 130	1	1	13	10

Anexos

	CU # 131	1	1	11	7
	CU # 132	1	1	9	2
	CU # 133	4	1	0	2
	CU # 134	4	1	0	3
	CU # 135	4	1	0	2
	CU # 136	1	1	10	13
	CU # 137	1	1	6	12
	CU # 138	1	1	3	13
	CU # 139	1	1	3	3
	CU # 140	1	1	1	12
	CU # 141	1	1	4	6
	CU # 142	1	1	5	6
	CU # 143	1	1	3	6
	CU # 144	1	1	4	11
	CU # 145	1	1	5	12
	CU # 146	1	1	3	9
	CU # 147	1	1	4	4
	CU # 148	2	1	7	2
	CU # 149	4	1	0	7

Anexos

	CU # 150	4	1	0	3
	CU # 151	3	1	0	2
	CU # 152	1	1	10	16
	CU # 153	1	1	6	13
	CU # 154	1	1	3	16
	CU # 155	1	1	3	3
	CU # 156	1	1	0	15
	CU # 157	1	1	4	5
	CU # 158	1	1	5	8
	CU # 159	1	1	3	5
	CU # 160	1	1	4	10
	CU # 161	1	1	5	12
	CU # 162	1	1	3	8
	CU # 163	1	1	3	4
	CU # 164	4	1	0	4
	CU # 165	4	1	0	4
	CU # 166	4	1	0	3
	CU # 167	4	1	0	2
	CU # 168	1	1	8	82

Anexos

	CU # 169	1	1	0	4
	CU # 170	1	1	1	15
	CU # 171	2	1	3	8
	CU # 172	1	1	0	2
	CU # 173	2	1	2	15
	CU # 174	1	1	3	22
	CU # 175	4	1	4	43
	CU # 176	2	1	3	25
	CU # 177	1	1	1	16
	CU # 178	1	1	1	7
	CU # 179	1	1	3	3
	CU # 180	1	1	1	2
	CU # 181	1	1	1	4
	CU # 182	2	1	3	8
	CU # 183	1	1	4	5
	CU # 184	2	1	2	20
	CU # 185	3	1	0	2
	CU # 186	1	1	0	0
	CU # 187	1	1	0	0

Anexos

	CU # 188	2	1	3	62
	CU # 189	3	1	1	4
	CU # 190	2	1	3	6
	CU # 191	1	1	4	6
	CU # 192	1	1	4	8
	CU # 193	1	1	2	12
	CU # 194	1	1	1	13
	CU # 195	1	1	2	18
	CU # 196	1	1	0	2
	CU # 197	1	1	19	41
	CU # 198	1	1	1	2
	CU # 199	1	1	3	15
	CU # 200	1	1	3	8
	CU # 201	1	1	4	19
	CU # 202	2	1	3	20
	CU # 203	4	1	4	16
	CU # 204	3	1	7	10
	CU # 205	3	1	8	14
	CU # 206	3	1	1	5

Anexos

	CU # 207	4	1	4	4
	CU # 208	2	1	3	10
	CU # 209	9	1	1	60
	CU # 210	1	1	1	38
	CU # 211	4	1	1	16
	CU # 212	3	1	1	10
	CU # 213	1	1	0	2
	CU # 214	1	1	1	1
	CU # 215	1	1	1	3
	CU # 216	3	1	1	98
	CU # 217	1	1	1	2
	CU # 218	3	1	1	44
	CU # 219	1	1	0	2
	CU # 220	1	1	0	2
	CU # 221	1	1	2	12
Registro y Notarias	CU # 1	1	1	1	3
	CU # 2	6	1	4	10
	CU # 3	1	1	3	5
	CU # 4	1	1	6	2

Anexos

	CU # 5	1	1	0	0
	CU # 6	1	1	1	12
	CU # 7	5	1	5	20
	CU # 8	1	1	0	5
	CU # 9	1	1	0	3
	CU # 10	1	1	2	15
	CU # 11	1	1	13	94
	CU # 12	2	1	11	80
	CU # 13	4	1	5	72
	CU # 14	3	1	3	54
	CU # 15	1	1	5	26
	CU # 16	1	1	1	8
	CU # 17	1	1	6	20
	CU # 18	1	1	0	2
	CU # 19	1	1	1	5
	CU # 20	1	1	0	71
	CU # 21	8	1	0	120
	CU # 22	13	1	2	104

Anexos

Anexo 5.

Encuesta para determinar la evaluación del procedimiento de estimación de tiempo y esfuerzo de las pruebas de liberación en el laboratorio de Calidad Central. Para ello necesitamos conocer el grado de dominio que usted posee sobre el proceso; y con ese fin deseamos que responda lo que se le pide a continuación.

Nombre y apellidos: _____

Labor que realiza: _____

Años de experiencia: _____ Especialidad: _____

Categoría docente: _____ Categoría científica: _____

- Marque con una (X) el grado de conocimiento que usted tiene sobre la temática que se investiga.

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

- Marque con una (X) las fuentes que le han servido para argumentar su conocimiento. De la temática que se investiga. Encierre en un círculo la que más ha influido.

No	Fuentes de Argumentación	Grado de influencia		
		Alto	Medio	Bajo
1	Análisis realizado por usted.			
2	Experiencia.			
3	Trabajo de autores nacionales.			
4	Trabajo de autores extranjeros.			
5	Su propio conocimiento del tema.			
6	Su intuición.			

Anexos

Cuestionario:

1. ¿Considera usted que los elementos (complejidad de los casos de uso y experiencia de los probadores) son suficientes para llevar a cabo un procedimiento de estimación de tiempo y esfuerzo en las pruebas de liberación?

Innecesarios.

Necesarios pero no suficientes para llevar a cabo un procedimiento de estimación de tiempo y esfuerzo.

Necesarios y suficientes para llevar a cabo un procedimiento de estimación de tiempo y esfuerzo.

Otras consideraciones al respecto:

2. Considera usted que los elementos que se tuvieron en cuenta en el procedimiento para clasificar la complejidad de los casos de uso (cantidad de flujos y elementos de entrada y/o salida) son :

Innecesarios.

Necesarios pero no suficientes para clasificar la complejidad de los casos de uso.

Necesarios y suficientes para clasificar la complejidad de los casos de uso.

Otras consideraciones al respecto:

3. Considera usted que los elementos que se tuvieron en cuenta en la gestión por competencia para clasificar la experiencia de los probadores son:

Innecesarios.

Necesarios pero no suficientes para clasificar la experiencia de los probadores.

Necesarios y suficientes para clasificar la experiencia de los probadores.

Anexos

Otras consideraciones al respecto:

4. ¿El procedimiento se ajusta al proceso de pruebas de liberación que se le aplica a los proyectos de la universidad en el laboratorio de Calidad Central?

Totalmente.
 En gran medida.
 Poca.

¿Por qué?

5. ¿Considera usted que el procedimiento propuesto resulta ser una mejora con respecto a la actual forma de realizar la estimación de tiempo y esfuerzo en el laboratorio de Calidad Central?

Totalmente.
 En gran medida.
 Poca.

¿Por qué?

6. ¿Considera usted que la aplicación del procedimiento propuesto puede ser efectivo en la planificación de las pruebas de liberación de un proyecto?

Totalmente.
 En gran medida.
 Poca.

Anexos

¿Por qué?

7. Elabore un comentario general sobre el procedimiento que está siendo evaluado.

8. ¿Qué recomendaciones usted aportaría para la mejora del procedimiento propuesto?

Anexo 6:

Caracterización de Expertos:

Experto	Categoría docente	Graduado	Labor que realiza	Años vinculados a la UCI
1	Máster	2000	Profesor Asistente de la UCI.	2
2	Instructor	2007	Jefa del Laboratorio Industrial de Pruebas de software.	7
3	Adiestrado	2007	Especialista de la Dirección de Calidad.	2

Anexos

4	Instructor	2005	Jefa del Departamento Central de IGSW.	3
5	Adiestrado	2007	Especialista de la Dirección de Calidad.	2
6	Asistente	2003	Especialista de calidad de software.	6
7	Instructor	2007	Especialista del Departamento de Métricas.	2
8	Instructor	2005	Profesor de Ingeniería de Software.	4

Glosario de Términos

Glosario de Términos

Aduana: Sistema Integral para la Gestión de Recursos Humanos.

Calidad de software: La concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados, y con las características implícitas que se espera de todo software desarrollado profesionalmente. [PRESSMAN, 2005].

CICPC: Contrato de la Solución Tecnológica Integral del Cuerpo de Investigaciones Científicas, Penales y Criminalísticas, adscrito al Ministerio del Poder Popular para Relaciones Interiores y Justicia de la República Bolivariana de Venezuela.

COCOMO: Es un modelo de estimación de costes de software. Fue desarrollado por Barry W. Boehm a finales de los 70 y comienzos de los 80.

Experto: El experto es más bien maestro de un conocimiento, que integra naturalmente elementos de conocimiento, pero que tiene en cuenta una experiencia y saberes transmitidos de modo no formalizado, el experto se define pues como el hombre capaz de emitir un juicio sobre un tema.

Lluvia de ideas: Este es un modelo que se usa para generar ideas. La intención en su aplicación es la de generar la máxima cantidad posible de requerimientos para el sistema. No hay que detenerse en pensar si la idea es o no del todo utilizable. La intención de este ejercicio es generar, en una primera instancia, muchas ideas. Luego, se irán eliminando en base a distintos criterios. (Chaves, 2007)

Prisiones: Proyecto para la Humanización Penitenciaria.

Procedimiento: Sucesión cronológica de operaciones concatenadas entre sí, que se constituyen en una unidad de función para la realización de una actividad o tarea específica dentro de un ámbito predeterminado de aplicación.

Proyecto: Es el elemento organizativo a través del cual se gestiona el desarrollo de software y el resultado del mismo es una versión de un producto.

Glosario de Términos

Riesgo: Un riesgo de un proyecto es un evento o condición inciertos que, si se produce, tiene un efecto positivo o negativo sobre al menos un objetivo del proyecto, como tiempo, coste, alcance o calidad.

Roger. S. Pressman: Es una autoridad internacionalmente reconocida en la mejora de proceso de software y en tecnología de Ingeniería de Software. Por más de tres décadas, ha trabajado como ingeniero, gerente, profesor, autor y consultor de software en temas de Ingeniería de Software. Actualmente es presidente de R. S. Pressman and Asóciate, Inc., una firma consultora especialista en métodos y entrenamiento en Ingeniería de Software.

RUP: El Rational Unified Process (RUP) es una metodología formal, a veces también llamada proceso. El RUP describe a gran detalle todas las actividades, roles, responsabilidades, productos de trabajo y herramientas para definir quién hace qué y en qué momento en un proyecto de desarrollo de software.

SIGESC: Sistema de Gestión de Emergencias de Seguridad Ciudadana.

Software: Conjunto de instrucciones que las computadoras emplean para manipular datos.

UCI: Universidad de las Ciencias Informáticas.