

“PROCESO DE PRUEBAS DE REQUISITOS BASADOS EN LA INGENIERÍA DE REQUISITOS PARA SER APLICADA A LOS PROYECTOS DE SOFTWARE”

Documento Proyecto de grado

Junio 09 de 2011

Juan Camilo Hernández González

John Felipe Morales Isaza

**UNIVERSIDAD EAFIT
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS**

MEDELLIN

2011

TABLA DE CONTENIDO

INTRODUCCIÓN	4
DEFINICIÓN DEL PROBLEMA	6
OBJETIVOS	9
OBJETIVO GENERAL	9
OBJETIVOS ESPECÍFICOS	9
ESTADO DEL ARTE	10
1. Peores bugs de la historia por falta de pruebas:	10
2. Ingeniería de Requisitos	12
3. Importancia de la Ingeniería de Requisitos	13
4. Roles involucrados en la Ingeniería de Requisitos	14
5. Principales modelos del proceso de Ingeniería de Requisitos	16
6. ¿Qué es un Requisito?	29
7. Diferencia entre Requerimiento vs Requisito	29
8. ¿Para qué son los requisitos?	30
9. Importancia de los requisitos	30
10. Clasificación de los Requisitos	31
11. Características de los Requisitos	33
12. Dificultades para definir los requisitos	35
13. Técnicas principales para obtener requisitos	36
14. Características de un analista de requisitos efectivo	38
ESTUDIO DE LOS PROCESOS ACTUALES DE PRUEBAS EN REQUISITOS EN LAS EMPRESAS.	40
1. Situación Actual	43
2. Las empresas de servicios en EEUU y Europa	45
3. Las empresas de servicios en Colombia	50
ESTUDIO DE LOS PROYECTOS Y TIPIFICACIÓN DE PRUEBAS EN REQUISITOS.	58
ADAPTACIÓN DEL PROCESO	61

1. Generalidades	61
2. Proceso propuesto de Pruebas en Requisitos.....	66
CONCLUSIONES	91
GRUPO DE TRABAJO	94
BIBLIOGRAFÍA	95

TABLA DE CONTENIDO DE IMÁGENES

1. Ariane 5 Vuelo 501	9
2. Acelerador médico Therac.....	10
3. Sonda Mariner	11
4. El ping de la muerte.....	12
5. Modelo Pohl.....	16
6. Modelo en Espiral.....	18
7. Modelo de Ingeniería de Requisitos por Durán, Bernárdez, Toro y Ruiz.....	19
8. Elicitación y Negociación.....	20
9. Análisis de Requisitos.....	20
10. Validación de Requisitos.....	21
11. Modelo SWEBOK.....	21
12. Modelo BABOK.....	24
13. Modelo RUP.....	25
14. Imagen iso9126.....	32
15. Situación actual.....	44
16. Situación actual2.....	45
17. Costo de corregir errores en las distintas fases del desarrollo.....	53
18. Métricas de un proyecto sobre pruebas en requisitos.....	60
19. Ciclo de vida de desarrollo de software.....	63
20. Alcance de los requisitos.....	65
21. Etapas de las Pruebas de Requisitos.....	66
22. Proceso propuesto de Pruebas en Requisitos.....	67

INTRODUCCIÓN

En la actualidad existen diversas metodologías para el desarrollo de software, y con el pasar de los años, la rama de Ingeniería de software ha introducido una serie de estándares y metodologías para medir y mejorar la calidad tanto del sistema a desarrollar o producto final, como del proceso de desarrollo en sí. Sin embargo, los estándares y técnicas de pruebas de software, en general, no presentan un grado de especificidad mediante el cual el aseguramiento de la calidad del producto sea tenido en cuenta en todas las etapas de desarrollo del software.

A nivel internacional existen organizaciones tales como: IEEE (Institute of Electrical and Electronics Engineers), ISO (International Organization for Standardization), CMU (Carnegie Mellon University), SEI (Software Engineering Institute), IIT (Illinois Institute of Technology), entre otras, que desarrollan modelos de evaluación y mejora en el desarrollo de software, como CMMI (Capability Maturity Model Integration) y TMMI (Testing Maturity Model Integration) como modelo de pruebas complementario del CMMI, que aportan estándares y buenas prácticas en el ámbito de las pruebas de software, ayudando así, a un aseguramiento de la calidad en los aplicativos a desarrollar. Adicionalmente, existe un organismo internacional que la industria del software comienza a reconocer es el ISTQB (International Software Testing Qualifications Board), el cual se encarga de realizar certificaciones de los probadores y de los procesos de pruebas usados al interior de una organización.

Así pues, la Ingeniería de Requisitos cumple un papel primordial en el proceso de producción de software, ya que se enfoca en una fase fundamental que es la definición de lo que se desea producir para satisfacer las necesidades de los clientes. Su principal tarea, consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema. Con la aplicación de estos factores o propiedades, se busca minimizar costos y errores en las fases tempranas del ciclo de vida de desarrollo de software.

Es claro, que muchos de los problemas de la ingeniería de software, provienen de la imprecisión en la especificación de requisitos y la indiferencia de algunas empresas hoy día que no prestan mayor importancia a la ejecución de dichas pruebas; pues la mayoría de

empresas, concentran sus esfuerzos en otro tipo de fases del ciclo de vida del desarrollo de software, como la corrección de defectos en etapas finales o posteriores a la implementación de software, debido a que se tiene la creencia que generar esfuerzos en etapas tempranas, implica mayor número de personas que realicen esta actividad, mayor coste y no saben cómo justificar este tiempo ante el cliente.

A menudo no se desarrolla lo que el cliente desea, pues se tienen que estipular nuevos requisitos y se deben hacer cambios al sistema, retrasando la entrega de éste e incrementando el coste del proyecto, lo que puede llevar a una empresa a pagar altos precios tanto en su imagen como en pólizas y demás acuerdos generados con el cliente al inicio del proyecto.

Además, algunas empresas piensan que a medida que se implementan las respectivas soluciones a los requisitos planteados, se irán realizando las correcciones necesarias para cumplir con las necesidades de los clientes. Lo que no contemplan, es que la corrección de un error es menos costosa en la medida que sean detectados sus defectos en fases tempranas del proceso de software y entre más avanzado esté el desarrollo, la corrección será más costosa.

Algunas empresas de testing y áreas dedicadas a la calidad del software, a menudo intentan utilizar metodologías diferentes, emplean recursos y realizan otro tipo de inversiones buscando garantizar la satisfacción del cliente y productos 100% libre de errores, pero sus subprocesos fallan continuamente y no se logran los objetivos propuestos, ya que en su mayoría sólo contemplan la fase posterior a la implementación de software.

Una de las principales dificultades que presentan las empresas de calidad y testing colombianas, es el desconocimiento de técnicas de pruebas en requisitos en los diferentes tipos de proyectos que manejan las organizaciones. Es por esta razón, que se ha convertido en el objetivo principal de este proyecto de investigación, definir un proceso de pruebas de requisitos, para la verificación y validación de requisitos funcionales y no funcionales partiendo de la ingeniería de requisitos. De esta manera, se busca mejorar los procesos y técnicas, las cuales podrán ofrecer a sus clientes un factor diferenciador y que le permita convertirse en una empresa más competitiva con los servicios que ofrecen.

DEFINICIÓN DEL PROBLEMA

Actualmente la tecnología está presente en todas las áreas del conocimiento, y es por esta razón, que debe garantizarse la confiabilidad de los datos y de la información producida. Es por esto, que los usuarios toman decisiones y acciones que afectan directamente los beneficios de su negocio.

Los errores inyectados en el software son muy importantes de detectar, pues es necesario que un aplicativo este sin errores ya que cada uno de ellos, puede causar daños o pérdidas de la información generada por dicho sistema, entre otras consecuencias de gran magnitud. La mayor parte de las compañías, sin importar sus líneas de negocio, dependen de la tecnología, todas sus contabilidades, sus ventas, sus ingresos y egresos están totalmente sistematizados y sacan indicadores para la toma de decisiones, si esta información esta errónea le puede causar a una compañía pérdidas de dinero, de imagen, de confianza dentro del mercado y hasta en algunos casos, estos errores pueden hasta causar la muerte de los usuarios. Para disminuir estos riesgos, durante varios años la ingeniería de software a presentado varios modelos y estándares a los desarrollares de software para realizar buenas especificaciones de requisitos, análisis, diseño implementación y mantenimiento de software, todas estas técnicas y procedimientos pueden minimizar los riesgos a posibles errores de un sistema, y adicionalmente, aumenta la probabilidad de terminar un proyecto dentro los tiempos estimados, presupuesto estimado y con calidad.

El problema radica en que las empresas no se han concientizado de la importancia y los beneficios de las pruebas de requisitos, pues esto evitará sobre costos de implementación y aportará un aseguramiento de la calidad en el aplicativo desarrollado. Por otra parte, otro problema que tienen las empresas de calidad, es que creen que solo con contar con una lista de chequeo, se puede asegurar que el requisito a desarrollar y su diseño, cuenta con la estructura adecuada, para así tener un aseguramiento de la calidad optimo y con una carta de aprobación hacia el cliente y éste últimos a su vez, valide que la información suministrada, cumpla con las necesidades expuestas al inicio del proyecto. Es importante, concientizar a las empresas locales, que así como las pruebas funcionales tienen su proceso completo, las pruebas de requisitos también deben contar con un procedimiento robusto y estructurado, ya que entre más tempranas se hagan dichas pruebas será mejor para el proyecto y se asegurará

un aplicativo con altos estándares de calidad. La experiencia nos dice que “entre más temprano se encuentre un error, los costo y tiempos de corrección de errores, serán menores”.

En palabras de F.P Brooks

“La parte más difícil de construir un sistema software es decidir que construir.... Ninguna otra parte del trabajo afecta más negativamente al sistema final si se realiza de manera incorrecta. Ninguna otra parte es más difícil de rectificar después.”¹

Esto significa que la especificación de requisitos es una labor sumamente importante y de gran atención y cuidado para un proyecto de software, esta labor la debe realizar una persona experta, ya que cualquier error de definición o escritura, podrá conllevar a consecuencias incosteables y de alto impacto en fases posteriores del desarrollo de software. La gran mayoría de las empresas hoy en día, ignoran la importancia de una buena labor de especificación de requisitos, y los beneficios que pueden aportar al éxito de sus proyectos, ya que no basta solo con una definición de las necesidades de sus clientes, sino también prestar gran atención a la verificación y validación de éstos, ya que los riesgos de encontrar grandes fallos en etapas posteriores del ciclo de vida de desarrollo de software se minimizan, así como también los altos costos en reprocesos y a veces reingeniería que pueden llegar al fracaso de los proyectos.

Según lo anterior, se puede percibir, que al no contar con un proceso desarrollado y estructurado de validación y verificación de requisitos, podrán verse afectados factores como tiempo, costo y calidad.

¹ <http://www.slideshare.net/emsandoval/generalidades-ingeniera-de-requisitos>

OBJETIVOS

OBJETIVO GENERAL

Definir y documentar un proceso de pruebas de requisitos para proyectos de desarrollo de software, en donde se garantice la calidad de los insumos iniciales (requisitos) en etapas tempranas del ciclo de vida de desarrollo de software

OBJETIVOS ESPECÍFICOS

- Conocer el estado actual del proceso de pruebas de requisitos en empresas Colombianas que realizan pruebas de requisitos, las fortalezas y debilidades del proceso empleado en la actualidad, además de las estrategias, metodologías y tecnología utilizada.
- Realizar un análisis del proceso de validación y verificación que se está llevando a cabo para pruebas de requisitos bajo características de proyectos.
- Proponer, analizar y definir un proceso, para la ejecución de pruebas de requisitos.

ESTADO DEL ARTE

1. Peores bugs de la historia por falta de pruebas:

Es importante conocer un poco sobre las consecuencias y hechos ocurridos a través del tiempo por falta de la aplicación de procesos de ingeniería de requisitos. Las razones para incurrir en este hecho son varias, como el desconocimiento de metodologías y modelos actuales para la verificación y validación de requisitos, como también la premura de realizar entregas tempranas al cliente, que incumplen con los niveles de calidad mínimos. Es por esto, que se describen a continuación, algunas muestras de las consecuencias registradas en la historia sobre fracasos en los proyectos de software por no emplear mecanismos de calidad en cada fase del ciclo de vida del desarrollo de software.

1.1. 04 de junio 1996 – Ariane 5 Vuelo 501. En la reutilización del software del Ariane 4 para el Ariane 5 se obvió que había hardware diferente, lo que provocó que el cohete estallara. Uno de los errores considerados más caros de la historia.



Ariane 5 Vuelo 501

El lanzamiento del vuelo 501 del Ariane 5, tuvo lugar el 4 de Junio de 1996, fue la primera prueba de vuelo del sistema desechable de lanzamiento del Ariane 5. No fue un éxito, la lanzadera se destruyó 37 segundos después del lanzamiento debido a un mal funcionamiento en el software de control.

El software del Ariane 5 reutilizó las especificaciones del Ariane 4, pero la trayectoria del Ariane 5 era considerablemente distinta y superaba el rango para el cual el código reutilizado había sido diseñado.

² Imagen tomada de www.blogdetecnologia.com

Las pruebas previas al vuelo nunca fueron ejecutadas sobre el código reajustado bajo condiciones de vuelo simuladas del Ariane 5, por lo tanto el error no fue descubierto antes del lanzamiento.

1.2. 1985-1987 – Acelerador médico Therac – 25. Un error en el tratamiento de radiación del dispositivo proporcionó dosis letales de radiación en varios centros médicos.



Acelerador médico

Un número de pacientes recibieron una dosis letal de rayos gamma debido a un fallo de software.

El Therac – 25 era una máquina para radiación terapéutica producida por la empresa Atomic Energy of Canada Limited.

Estuvo involucrada con al menos seis accidentes conocidos entre 1985 y 1987, en los cuales los pacientes fueron objetos de una sobredosis masiva de radiación.

Al menos cinco pacientes murieron por sobredosis y muchas otras se vieron afectadas. Estos accidentes destacan los riesgos del control software de sistemas críticos en términos de seguridad, lo que implica un mayor reconocimiento en la actualidad de una buena especificación y pruebas sobre los requisitos elicitados, ya que se puede determinar la completitud y exactitud sobre los requisitos a implementar.

1.3. 28 de julio 1962 – Mariner I. Un error en el software de la sonda Mariner I, y la falta de pruebas para detectar este error hace que el cohete se desvíe de su trayectoria cayendo sobre el Océano Atlántico.

³ Imagen tomada de www.andaluciainvestiga.com, 01/02/11 – 20:00



Mariner

1.4. Noviembre de 2000 – National Cancer Institute, Ciudad de Panamá. El software de Multidata Systems International, una empresa de EE.UU, no calculó la dosis correcta de radiación para los pacientes sometidos a radioterapia esto se debe a que desde el principio en la parte de elicitación se indicó mal la dosis apropiada. Ocho personas murieron.

Luego de conocer y entender un poco las consecuencias de no adaptar un proceso de verificación y validación de requisitos, se ahondará un poco sobre la temática de este proyecto de grado, el cual estará enfocado a la ingeniería de requisitos y aún más exactamente, en las pruebas que pueden realizarse sobre los requisitos

2. Ingeniería de Requisitos

Para mayor conocimiento del lector, se indicarán algunas definiciones de ingeniería de requisitos, generadas por algunos reconocidos autores:

- "Ingeniería de Requisitos es la disciplina para desarrollar una especificación completa, consistente y no ambigua, la cual servirá como base para acuerdos comunes entre todas las partes involucradas y en dónde se describen las funciones que realizará el sistema". (**Boehm - 1979**).
- "Ingeniería de Requisitos es el proceso por el cual se transforman los requisitos declarados por los clientes, ya sean hablados o escritos, a especificaciones precisas, no

⁴ Imagen tomada de http://es.wikipedia.org/wiki/Mariner_1, 01/02/11 – 20:00

ambiguas, consistentes y completas del comportamiento del sistema, incluyendo funciones, interfaces, rendimiento y limitaciones". **(STARTS Guide - 1987)**.

- "Es el proceso mediante el cual se intercambian diferentes puntos de vista para recopilar y modelar lo que el sistema va a realizar. Este proceso utiliza una combinación de métodos, herramientas y actores, cuyo producto es un modelo del cual se genera un documento de Requisitos" **(Leite - 1987)**.
- "Ingeniería de Requisitos es un enfoque sistémico para recolectar, organizar y documentar los Requisitos del sistema; es también el proceso que establece y mantiene acuerdos sobre los cambios de Requisitos, entre los clientes y el equipo del proyecto" **(Rational Software)**
- "El proceso sistemático de desarrollar requisitos mediante un proceso iterativo y cooperativo de analizar el problema, documentar las observaciones resultantes en varios formatos de representación y comprobar la precisión del conocimiento obtenido." **(CHRISTEL Y KANG - 1992)**
- "La Ingeniería de Requisitos es el proceso de estudiar las necesidades del usuario para llegar a una definición de requisitos de sistema, hardware o software." **(IEEE - 1990)**
- "La ingeniería de requisitos puede considerarse como un proceso de descubrimiento y comunicación de las necesidades de clientes y usuarios y la gestión de los cambios de dichas necesidades." **(AMADOR DURAN - 2000)**

3. **Importancia de la Ingeniería de Requisitos (IR)**

"Al aplicar la ingeniería de requisitos de manera adecuada, trae varios beneficios, como se muestran a continuación:

- Permite gestionar las necesidades del proyecto en forma estructurada: Cada actividad de la IR consiste de una serie de pasos organizados y bien definidos.

- Mejora la capacidad de elaborar cronogramas de proyectos, así como sus resultados: La IR proporciona un punto de partida para controles subsecuentes y actividades de mantenimiento, tales como estimación de costos, tiempo y recursos necesarios.
- Disminuye los costos y retrasos del proyecto: Muchos estudios han demostrado que reparar errores por un mal desarrollo no descubierto a tiempo, es sumamente costoso.
- Mejora la calidad del software: La calidad en el software tiene que ver con cumplir un conjunto de Requisitos (funcionalidad, facilidad de uso, confiabilidad, desempeño, entre otras).
- Mejora la comunicación entre equipos: La especificación de Requisitos representa una forma de consenso entre clientes y desarrolladores. Si este consenso no ocurre, el proyecto no será exitoso.
- Evita rechazos de usuarios finales: La ingeniería de Requisitos obliga al cliente a considerar sus Requisitos cuidadosamente y revisarlos dentro del marco del problema, por lo que se le involucra durante todo el desarrollo del proyecto”⁵.

4. Roles involucrados en la Ingeniería de Requisitos

Dentro de un proceso de ingeniería de requisitos son involucrados muchos actores esto es importante porque cada uno tienen diferentes actividades, se tiene que tener muy claro que cada uno de estos actores tienen responsabilidades y actividades muy diferentes dentro de un proyecto de software, el conocimientos de cada rol es importante y aporta a cada fase del ciclo de vida de desarrollo.

Hacer caso omiso a los intereses de las personas que participarán en el proyecto, puede acarrear inconvenientes de comunicación entre clientes y desarrolladores, inconvenientes que pueden ocasionar impactos en cuanto al presupuesto del proyecto y el tiempo.

⁵ Tomado de Ingeniería De Requisitos Ingeniería De Software,
<http://www.monografias.com/trabajos6/resof/resof.shtml>, 08/12/10 – 21:00

Los roles más importantes pueden clasificarse de la siguiente manera:

- “Usuario final: Son las personas que usarán el sistema desarrollado. Ellos están relacionados con la usabilidad, la disponibilidad y la fiabilidad del sistema; están familiarizados con los procesos específicos que debe realizar el software, dentro de los parámetros de su ambiente laboral. Serán quienes utilicen las interfaces y los manuales de usuario.
- Usuario Líder: Son los individuos que comprenden el ambiente del sistema o el dominio del problema en donde será empleado el software desarrollado. Ellos proporcionan al equipo técnico los detalles y Requisitos de las interfaces del sistema.
- Personal de Mantenimiento: Para proyectos que requieran un mantenimiento eventual, estas personas son las responsables de la administración de cambios, de la implementación y resolución de anomalías. Su trabajo consiste en revisar y mejorar los procesos del producto ya finalizado.
- Analistas y programadores: Son los responsables del desarrollo del producto en sí; ellos interactúan directamente con el cliente.
- Personal de pruebas: Se encargan de elaborar y ejecutar el plan de pruebas para asegurar que las condiciones presentadas por el sistema son las adecuadas. Son quienes van a validar si los Requisitos satisfacen las necesidades del cliente”.⁶

Otras personas que pueden estar involucradas, dependiendo de la magnitud del proyecto, pueden ser: administradores de proyecto, documentadores, diseñadores de base de datos, entre otros.”⁷

⁶ tomado de Ingeniería De Requisitos Ingeniería De Software,
<http://www.monografias.com/trabajos6/resof/resof.shtml>, 08/12/10 – 21:00

⁷ www.monografias.com > Ingeniería 01/02/11 – 20:00

5. Principales modelos del proceso de Ingeniería de Requisitos

A continuación se describen algunos de los principales modelos a nivel mundial de los procesos de Ingeniería de Requisitos:

5.1. Modelo Pohl



modelo Pohl

En el proceso de Ingeniería de Requisitos del modelo Pohl podemos identificar y extraer cuatro actividades principales, las cuales son:

5.1.1. Elicitación:

Consiste en hacer explícito el conocimiento oculto sobre las necesidades de clientes y usuarios y el sistema a desarrollar de forma que todos los participantes en el problema sean capaces de entenderlo.

Tácitamente se asume, que durante la realización de las actividades de elicitación es necesario:

- Identificar las fuentes de información.
- Conocer de la mejor forma posible el dominio del problema.
- Reutilizar especificaciones de requisitos similares.

⁸ Tomado del Curso de Ingeniería de Requisitos, dictado en Febrero 5 – Marzo 19 de 2011, por el Ingeniero Faber López. 10/02/11 – 08:00

- Utilizar las técnicas habituales de elicitación (entrevistas, casos de uso, cuestionarios, prototipos, lluvia de ideas, entre otras)

5.1.2. Negociación:

Tiene como objetivo lograr acuerdos entre todos los participantes sobre los requisitos elicitados.

Pohl propone tener en cuenta los siguientes factores:

- Explicitar los conflictos y evitar los conflictos emocionales entre los participantes. De esta forma, queda claro qué es lo que se negocia y que la negociación no esté afectada por motivos personales.
- Explicitar para cada conflicto las alternativas, las argumentaciones y las razones que los provocan. Así, la negociación se puede basar en las raíces del conflicto.
- Asegurar de que se toman las decisiones correctas. De este modo, los participantes estarán de acuerdo con los resultados de la negociación y no se sentirán desplazados por el proceso.
- Asegurar la participación de las personas adecuadas en el momento adecuado. Se evita con esto tener que volver a replantear las negociaciones, al no haber participado en las negociaciones alguno de los afectados.

5.1.3. Especificación / Documentación:

El objetivo es claro: documentar los requisitos elicitados y negociados.

5.1.4. Validación y Verificación:

El objetivo es comprobar que los requisitos documentados corresponden a las necesidades de los clientes y usuarios (validación) y comprobar que la especificación cumple los criterios de calidad oportunos (verificación).

5.2. “Modelo en Espiral”



El modelo en espiral fue planteado por Boehm para la ingeniería de requisitos (Boehm1994) y el Inquiry – Based Model de Potts (Potts1994).

En el proceso de Ingeniería de Requisitos del modelo en espiral se pueden identificar y extraer tres actividades principales que son:

5.2.1 Elicitación:

En esta actividad, se consultan las diferentes fuentes de información como clientes, usuarios, expertos en dominio, etc., con el propósito de entender el dominio del problema y establecer los requisitos del sistema a desarrollar.

En esta actividad es factible que los requisitos elicitados no sean completos y que se expresen en forma vaga o no estructurada

5.2.2 Análisis y Validación:

Realizar la integración y análisis de los requisitos elicitados.

Producto de esto, se pueden identificar los requisitos que faltan, las inconsistencias que se presenten y los conflictos que se puedan dar entre los requisitos.

Sawyer (1997) plantea la actividad de análisis y validación de requisitos, pero no hace ninguna referencia a la actividad de validación como tal, esto es, que haya confirmación por parte de clientes y usuarios de que los requisitos expresan sus necesidades y especifican el sistema que desean.

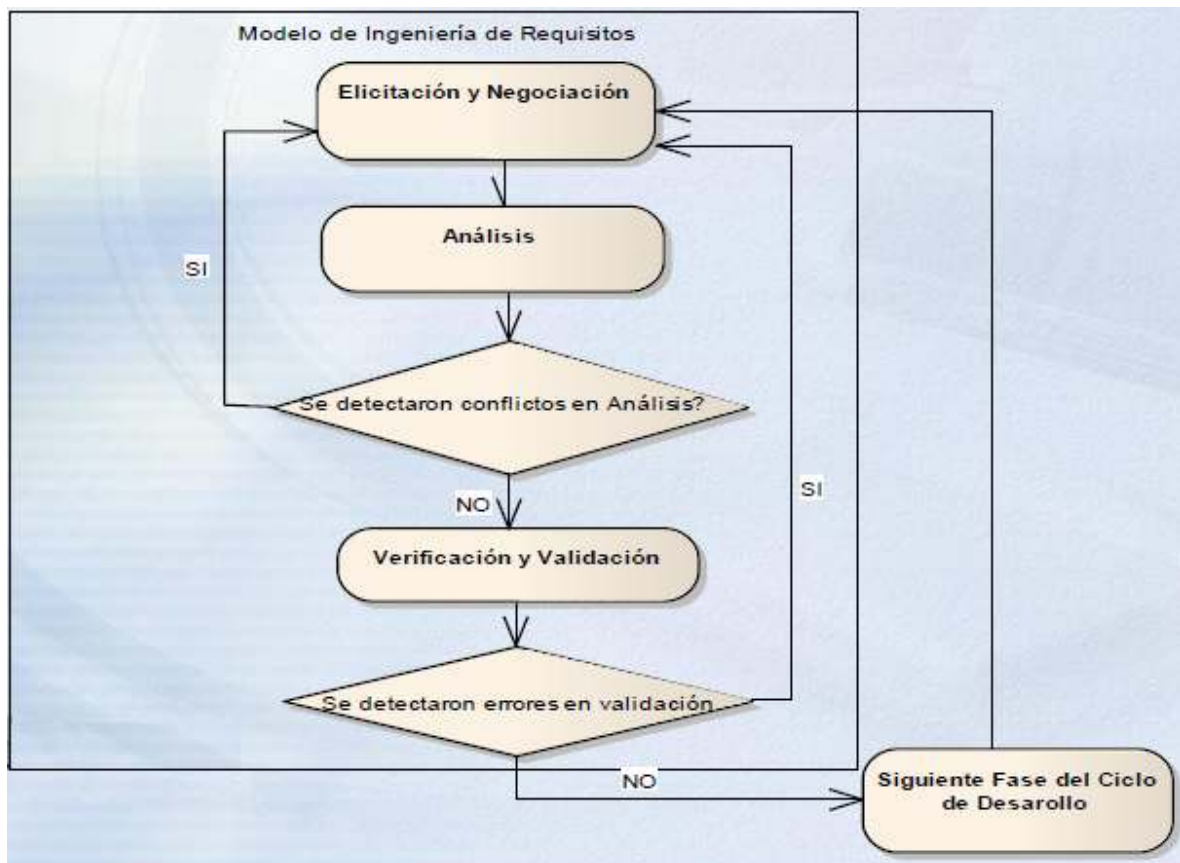
⁹ Tomado del Curso de Ingeniería de Requisitos, dictado en Febrero 5 – Marzo 19 de 2011, por el Ingeniero Faber López. 10/02/11 – 08:00

5.2.3 Negociación:

Realizar la integración y análisis de los objetivos es solucionar todos los conflictos que se hayan presentado durante el análisis, llegando a acuerdos entre los participantes en el proceso.

Normalmente hay que elicitar nuevamente (Incremental)."

5.3. "Modelo de Ingeniería de Requisitos por Durán, Bernárdez, Toro y Ruiz – 1999



Modelo de Ingeniería de Requisitos¹⁰

En el proceso de Ingeniería de Requisitos del modelo en espiral se pueden identificar y extraer tres actividades principales que son:

¹⁰ Tomado del Curso de Ingeniería de Requisitos, dictado en Febrero 5 – Marzo 19 de 2011, por el Ingeniero Faber López. 23/03/11 – 20:00

5.3.1. Elicitación y Negociación:



Elicitación y negociación¹¹

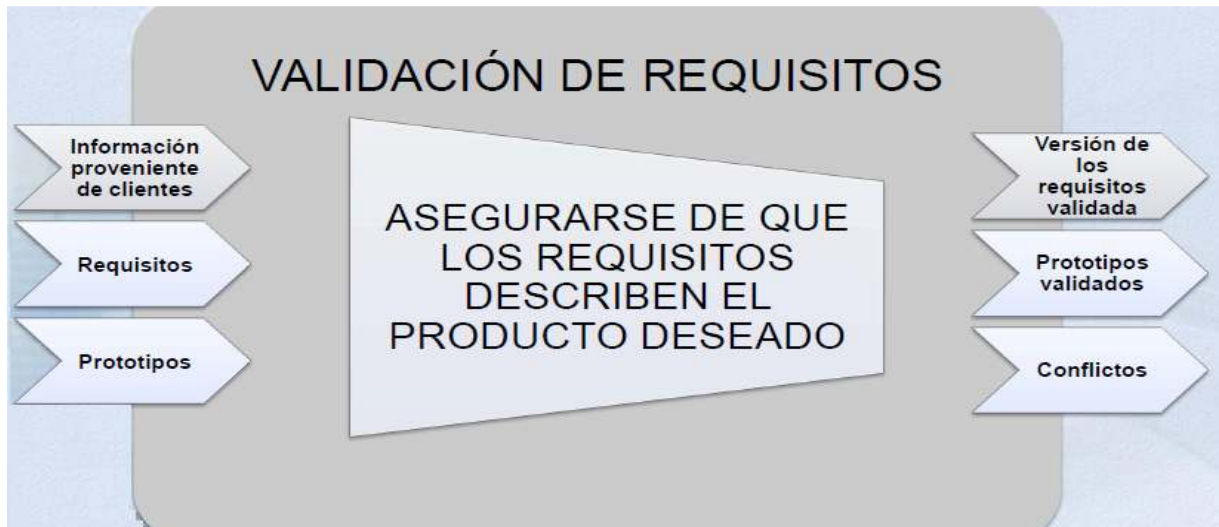
5.3.2. Análisis de Requisitos:



Análisis de requisitos

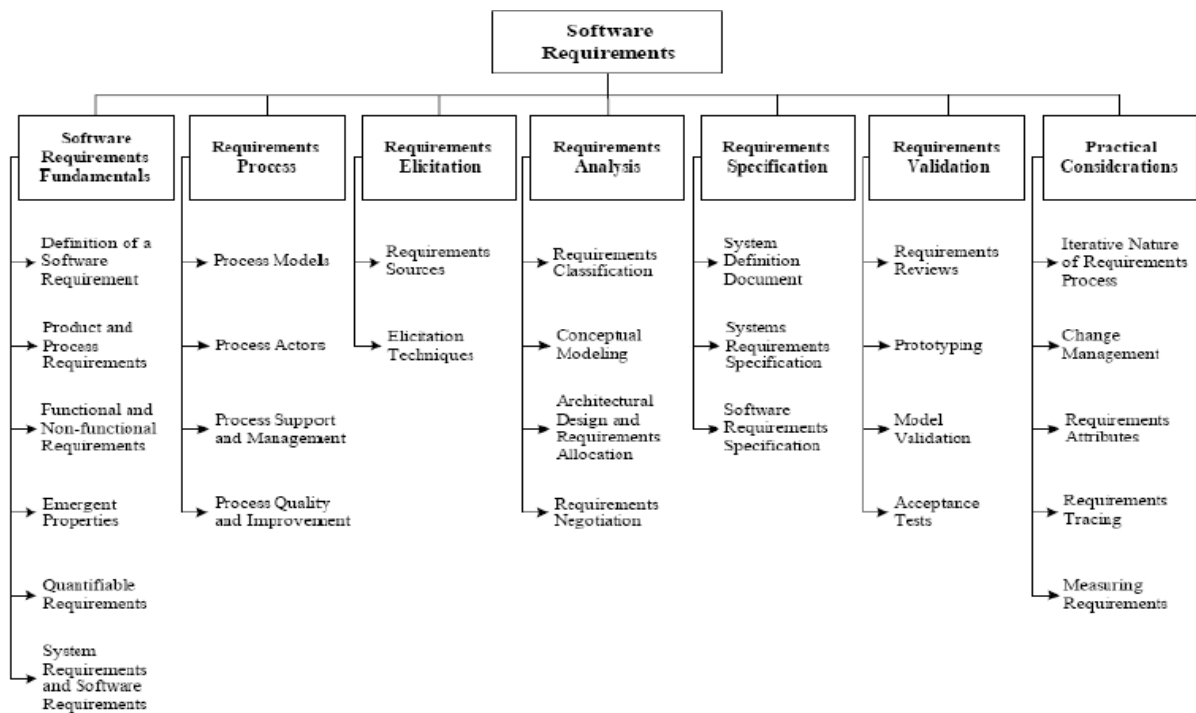
¹¹ Tomado del Curso de Ingeniería de Requisitos, dictado en Febrero 5 – Marzo 19 de 2011, por el Ingeniero Faber López.

5.3.3. Validación de Requisitos:



Validación de requisitos

5.4. Modelo SWEBOK



Etapas del modelo swebok ¹²

¹² Tomado del Curso de Ingeniería de Requisitos, dictado en Febrero 5 – Marzo 19 de 2011, por el Ingeniero Faber López. 23/03/11 – 20:00

SWEBOK recoge conceptos previos de la IEEE (IEEE Std 1233 - 1998) y otros trabajos en el que destacan el de Ian Sommerville, lo que trae como resultado una estructura más acorde a las necesidades de las empresas de desarrollo para la definición de un proceso de requisitos, entre las que se destacan conceptos y temas relacionados con: Conocimiento del Dominio, Clasificación de Requisitos, Identificación de fuentes de Requisitos, Técnicas de Elicitación, Análisis de requisitos, Validación de requisitos, Rastreabilidad y Gestión del Cambio

En el proceso de Ingeniería de Requisitos del modelo SWEBOK se pueden identificar y extraer siete actividades principales que son:

5.4.1. Fundamentos de Requisitos Software:

Se realiza una definición específica de las necesidades concernientes a problemas del mundo real que deben ser direccionados por un software.

El modelo SWEBOK propone la siguiente categorización de requisitos:

- Requisitos de producto y de proceso.
- Requisitos Funcionales y No Funcionales.
- Propiedades Emergentes
- Requisitos de Sistema y de Software

5.4.2. Proceso de Requisitos:

Introduce los conceptos de la definición del proceso necesario para ejecutar las labores de requisitos definiendo:

- Modelo de Procesos
- Actores del Proceso
- Procesos de Soporte y de Gestión
- Procesos de Mejora y Calidad

5.4.3. Elicitación de Requisitos:

Define los conceptos necesarios para identificar el origen de los requisitos y los mecanismos necesarios para su definición y recolección, donde se hace referencia a técnicas comunes para cumplir con esta actividad por lo cual es necesario definir:

- Fuentes de Requisitos
- Técnicas de Elicitación

5.4.4. Análisis de Requisitos:

El análisis de requisitos tiene como objetivo:

- Detectar y resolver conflictos entre requisitos
- Definir los límites del sistema y como éste interactúa con el ambiente
- Elaborar los requisitos del sistema y derivar los del software

Para cumplir los anteriores objetivos del análisis de requisitos, se deben seguir los siguientes pasos:

- Clasificar los Requisitos
- Generar modelo conceptual
- Generar Diseño Arquitectónico y asignar los requisitos
- Negociar los requisitos

5.4.5. Especificación de Requisitos:

En el ámbito de la ingeniería de software se refiere a la generación de un documento, el cual es sistemáticamente revisado, evaluado y aprobado. Dependiendo del tipo de sistema puede contener especificaciones de componentes no software. Pueden ser generados tres tipos de especificaciones:

- Documento de definición del sistema
- Especificación de requisitos del sistema
- Especificación de requisitos software

5.4.6. Validación de Requisitos:

Este será el punto central de análisis de este proyecto de grado, en donde los documentos de requisitos pueden estar sujetos a procedimientos de verificación y validación, de manera que se garantice el proceso para generarlos, el entendimiento, consistencia y completitud de los mismos. Para esto es posible aplicar diferentes tipos de técnicas como:

- Revisiones de requisitos
- Prototipos
- Modelos de Validación
- Pruebas de aceptación

5.4.7. Consideraciones Prácticas:

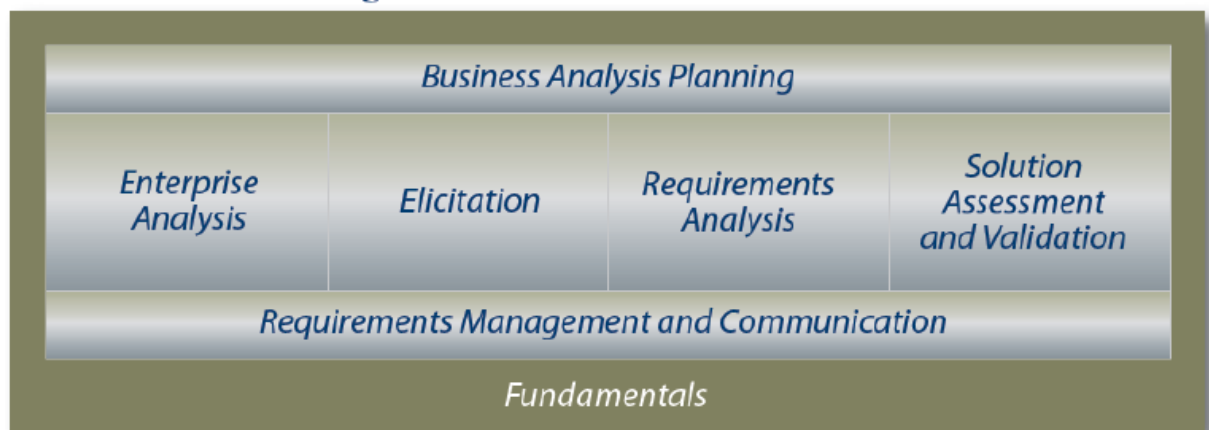
Así como el proceso de requisitos en un proyecto puede ser en sí mismo un ciclo completo de software, por lo cual es necesario considerar actividades no solo de recolección (elicitación) y verificación, sino también de soporte que permitan mejorar la calidad de los requisitos en etapas tempranas del proyecto para evitar así altos riesgos de implementación a futuro.

El modelo SWEBOK considera tener en cuenta las siguientes actividades de soporte:

- Gestión del Cambio
- Rastreabilidad de requisitos
- Mediciones

5.5. Modelo BABOK

BABOK™ v.2 Knowledge Areas



Modelo Babok¹³

En el proceso de Ingeniería de Requisitos del modelo BABOK se pueden identificar y extraer seis actividades principales que son:

5.5.1. Planeación y Monitoreo:

Describe cómo se determina, que actividades son necesarias y el orden de ejecución para un completo esfuerzo de análisis de negocio. Este cubre la identificación de los stakeholders, selección de las técnicas de elicitación, el proceso que debe ser seguido

¹³ Tomado del Curso de Ingeniería de Requisitos, dictado en Febrero 5 – Marzo 19 de 2011, por el Ingeniero Faber López.

para la gestión de requisitos y como se valorara el progreso del trabajo realizado y los cambios necesarios.

5.5.2. Análisis Empresarial:

Precisa cómo se deben tomar las necesidades de negocio, refinar y clarificar las que son necesarias y el alcance de la solución que puede ser implementada por el negocio. Cubre la definición y análisis, desarrollo del caso de negocio, estudios de factibilidad y la definición del alcance de negocio.

5.5.3. Elicitación:

Describe el trabajo que debe ser realizado con los stakeholders con el fin de reconocer sus necesidades y asegurar que con correctas y completamente entendidas.

5.5.4. Análisis de Requisitos:

Define cómo progresivamente se elabora la definición de la solución, para permitir al equipo del proyecto diseñar y construir una solución, que satisface las necesidades del negocio y de los stakeholders.

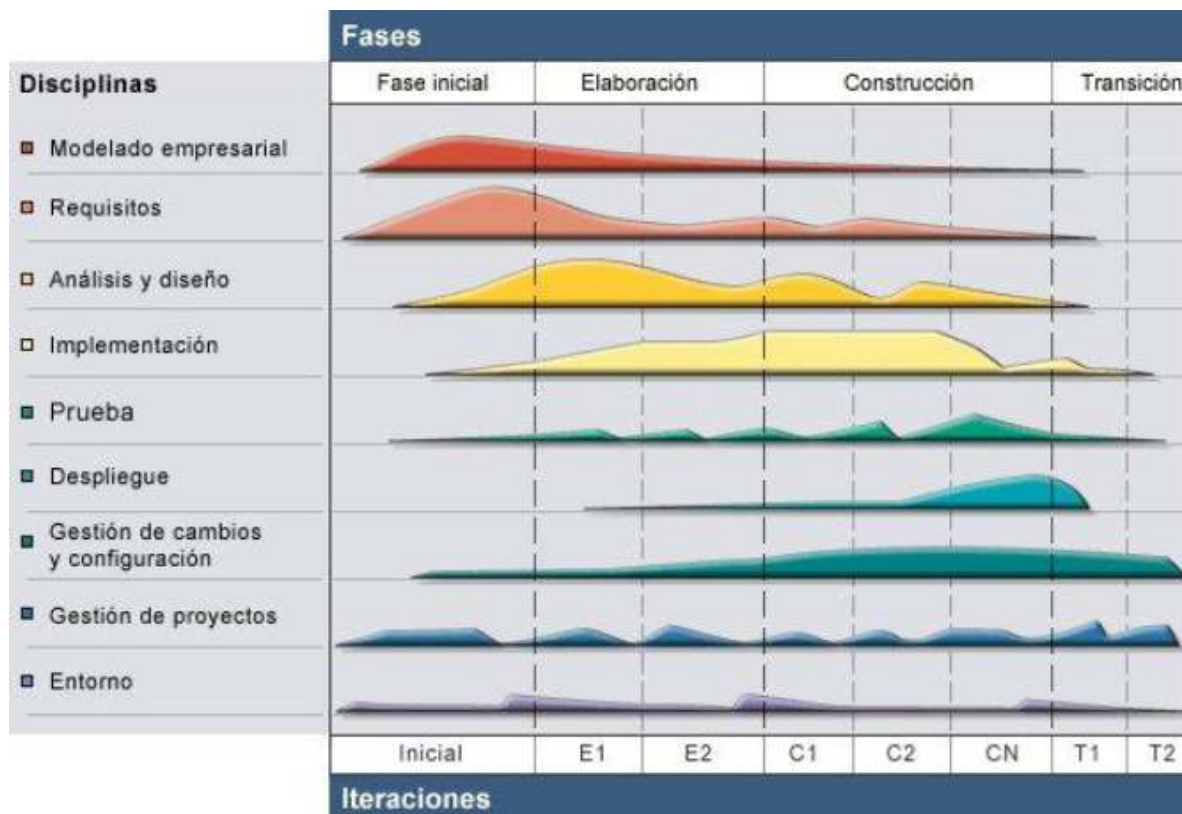
5.5.5. Evaluación y validación de la solución:

Describe como se evalúa la solución propuesta para determinar que es la que mejor se ajusta a las necesidades de negocio, e identifica posibles falencias en dicha solución de manera que se puedan visualizar los cambios necesarios para continuar con el proceso.

5.5.6. Gestión y comunicación de los Requisitos

Describe como se manejan los conflictos, eventos y cambios, y asegura que los stakeholders y el equipo del proyecto mantengan claro el acuerdo sobre el alcance de la solución.

5.6. Modelo RUP



Modelo RUP¹⁴

Esta disciplina explica, cómo obtener las solicitudes de los interesados y transformarlas en un conjunto de productos de trabajo de los requisitos que cubran el ámbito del sistema que va a crearse, y proporcionen requisitos detallados sobre lo que el sistema debe hacer.

En el proceso de Ingeniería de Requisitos del modelo RUP se pueden identificar y extraer seis actividades principales que son:

5.6.1. Analizar el problema:

La finalidad de esta actividad es llegar a un acuerdo sobre el problema que se va a resolver. El análisis del problema incluye la identificación de los implicados, la definición de los límites del sistema y la identificación de las restricciones impuestas al sistema.

¹⁴ Tomado del Curso de Ingeniería de Requisitos, dictado en Febrero 5 – Marzo 19 de 2011, por el Ingeniero Faber López. 23/03/11 – 20:00

5.6.2. Conocer las necesidades del interesado:

La finalidad de esta actividad es comprender las necesidades de los principales interesados reuniendo información acerca del producto deseado o concebido.

5.6.3. Definir el sistema:

La finalidad de esta actividad es empezar a converger en el ámbito de los requisitos de alto nivel haciendo un esquema de la extensión de los requisitos detallados del sistema.

5.6.4. Gestionar el ámbito del sistema:

La finalidad de esta actividad es conseguir que el ámbito del sistema que se está desarrollando sea lo más explícito posible, y se centre en un cuerpo gestionable de trabajo de requisitos para un ciclo de desarrollo.

5.6.5. Perfeccionar la definición del sistema:

La finalidad de esta actividad es perfeccionar aún más los requisitos para llegar a un consenso sobre la forma de entender la definición del sistema.

5.6.6. Gestionar cambios de requisitos:

La finalidad de esta actividad es evaluar el impacto de los cambios solicitados en los requisitos, y gestionar el impacto en sentido descendente de los cambios cuya implementación se ha aprobado.

5.7. Resumen de Actividades de la Ingeniería de Requisitos para diferentes modelos de procesos de Ingeniería de Software

En esta tabla se describen las actividades de cada modelo tiene para la ingeniería de requisitos donde se puede hacer un análisis que modelo es más efectivo y más completo para los proyectos de software.

MODELO	Oliver and Steiner 1996	EIA / IS-632	IEEE Std 1220- 1994	CMM nivel Repetitivo (2)	RUP
Actividades	Evaluar la información disponible	Análisis de requisitos	Análisis de requisitos	Identificación de requisitos	Análisis del Problema
	Definir métricas efectivas	Análisis funcional	Estudio de los requisitos	Identificación de restricciones del sistema a desarrollar	Comprender las necesidades de los involucrados
	Crear un modelo del comportamiento del sistema	Síntesis	Validación de requisitos	Análisis de los requisitos	Definir el sistema
	Crear un modelo de los objetos	Análisis y control del sistema	Análisis funcional	Representación de los requisitos	Analizar el alcance del proyecto
	Ejecutar el análisis		Evaluación y estudio de funciones	Comunicación de los requisitos	Modificar la definición del sistema
	Crear un plan secuencial de construcción y pruebas		Verificación de funciones	Validación de requisitos	Administrar los cambios de requisitos
			Síntesis		
			Estudio y evaluación del diseño		
			Verificación física		

			Control		
--	--	--	---------	--	--

Actividades de la ingeniería de requisitos¹⁵

6. ¿Qué es un Requisito?

A continuación presentaremos algunas definiciones de diferentes fuentes:

- “Un requisito es una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo.” (**IEEE Std 610.12 - 1990**)
- “Un requisito es una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal.” (**IEEE Std 610.12 - 1990**)
- “Un requisito es un atributo necesario dentro de un sistema, que puede representar una capacidad, una característica o un factor de calidad del sistema de tal manera que le sea útil a los clientes o a los usuarios finales.” (**IEEE Std 610.12 - 1990**)
- “Un requisito de software es una propiedad la cual debe ser exhibida con el fin de resolver algún problema en el mundo real.” (**SWEBOK**)

7. Diferencia entre Requerimiento vs. Requisito

A continuación se muestran algunas diferencias entre requerimiento y requisito:

Requerimientos	Requisitos
El requerimiento se refiere a la petición que se hace de algo, se requiere o solicita.	El requisito es la condición que debe cumplir algo, en general el requisito cumple con lo que se requiere o con el requerimiento (la solicitud).
El requerimiento estaría más relacionado en	El requisito estaría más relacionado en inglés

¹⁵ Tomado de Ingeniería De Requisitos Ingeniería De Software,
<http://www.monografias.com/trabajos6/resof/resof.shtml>, 25/03/10 – 21:30

inglés con la palabra <i>"request"</i> .	con la palabra <i>"requirement"</i> .
El requerimiento nos indica que debe hacer el sistema.	El requisito nos indica cómo hacer el software.
El requerimiento nos indica cómo resolver un problema o lograr un objetivo.	El requisito nos indica las condiciones que deben cumplirse para la resolución del problema.

8. ¿Para qué son los requisitos?

Los principales objetivos que se identifican en la especificación de requisitos de software son:

- Ayudar a los clientes a especificar lo que desean obtener mediante un software o aplicativo para satisfacer sus necesidades.
- Ayudar a los desarrolladores a entender qué quiere exactamente el cliente y/o usuario, pues son los requisitos la base fija con la que pueden trabajar.
Si no se realiza una buena especificación de requisitos, los costes de desarrollo pueden incrementarse considerablemente, ya que se deben hacer cambios durante la creación de la aplicación.

9. Importancia de los requisitos

- El éxito o fracaso de un proyecto está altamente influenciado por la calidad de los requisitos, y el proceso para gestionarlos durante el desarrollo de un producto.
- Establecen el alcance del trabajo subsecuente, pueden definir estrategia de desarrollo, riesgos, tomar decisiones de negocio (viabilidad de negocio), de proyecto (tiempo, recursos), de sistema (Arquitectura).
- Le indican al equipo del proyecto qué requieren los usuarios (necesidades de negocio).
- El no identificar los requisitos de manera adecuada, no permitirá minimizar la diferencia entre las necesidades de negocio y los requisitos del producto.

10. Clasificación de los Requisitos

Los requisitos pueden dividirse en funcionales, No funcionales, De información y Reglas de negocio:

- 10.1. “Los requisitos Funcionales** definen las funciones que el sistema será capaz de realizar. Describen las transformaciones que el sistema realiza sobre las entradas para producir salidas.

Atributos Funcionales de calidad según la ISO /ICE 9126:

- Adecuación (Suitability)
- Precisión (Accuracy): La exactitud se refiere a la precisión matemática de los cálculos hechos por el sistema. Cualquier sistema que realice cálculos numéricos debe considerar la exactitud.
- Conformidad (Compliance)
- Interoperabilidad (Interoperability): Es la medida de la habilidad de que un grupo de partes del sistema trabajen con otro sistema. Es un tipo especial de integrabilidad.
- Seguridad (Security): El sistema debe permitir ser usado solo por usuarios autorizados, y restringir el uso basado en permisos. El sistema no debe permitir a los usuarios omitir las reglas de seguridad o utilizar agujeros en la seguridad.

- 10.2. Los requisitos No Funcionales** tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares, etc.”¹⁶

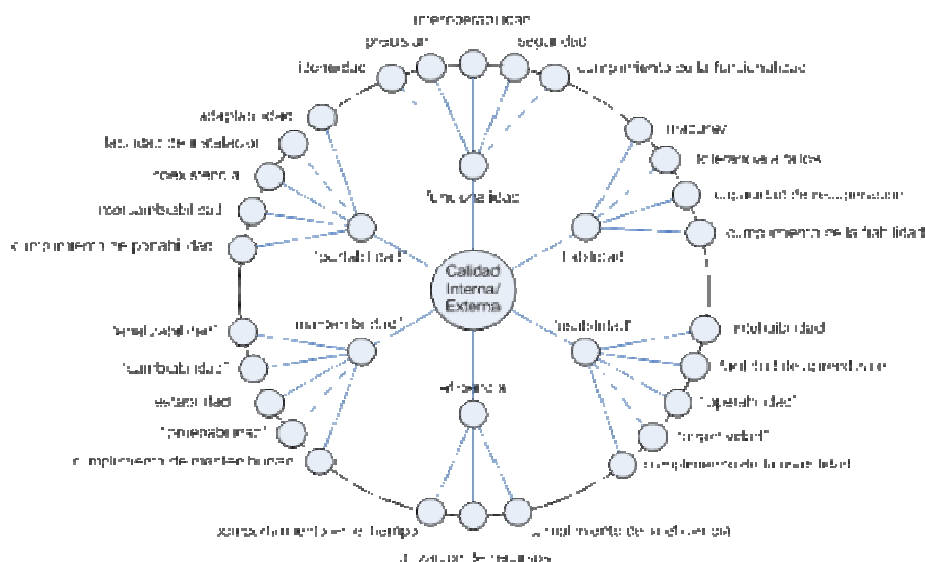
Atributos No Funcionales de Calidad según la ISO /ICE 9126:

- Fiabilidad (Reliability): Un conjunto de atributos relacionados con la capacidad del software de mantener su nivel de prestación bajo condiciones

¹⁶ Tomado de: <http://www.mitecnologico.com/Main/ProcesosDeLaIngenieriaDeRequerimientos> , 23/03/11 – 20:00
1. Ingeniería De Requisitos Ingeniería De Software,
<http://www.monografias.com/trabajos6/resof/resof.shtml>, 10/01/1 – 17:00
2. Monzón Antonio. Calidad de la Especificación. Documento: Calidad_de_la_Especificacion.pdf
3. Curso de Certificación - ISTQB

establecidas durante un período establecido (Madurez, Recuperabilidad y Tolerancia a fallos)

- Usabilidad (Usability): Un conjunto de atributos relacionados con el esfuerzo necesario para su uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios (Aprendizaje, Comprensión, Operatividad y Atractividad)
- Eficiencia (Efficiency): Conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesitados bajo condiciones establecidas (Comportamiento en el tiempo y Comportamiento de recursos)
- Mantenibilidad (Maintainability): Conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema software (Estabilidad, Facilidad de análisis, Facilidad de cambio y Facilidad de pruebas)
- Portabilidad (Portability): Conjunto de atributos relacionados con la capacidad de un sistema software para ser transferido desde una plataforma a otra (Capacidad de instalación, Capacidad de reemplazamiento, Adaptabilidad y Co-Existencia)



Características de la Calidad según la ISO/IEC 9126.¹⁷

¹⁷ Figura tomada de Fermín Figueroa, 25 de noviembre de 2009, NORMA ISO 9126, <http://normaiso9126.blogspot.com/> 23/03/11 – 20:00

10.3. Las Reglas de Negocios son todas aquellas restricciones, leyes, límites y cualquier otra condición que puedan restringir, condicionar o afectar la operación de un requisito funcional.

10.4. Los requisitos de Información son cada uno de los datos que serán almacenados, presentados, convertidos y eliminados por la ejecución de un requisito funcional.

11. Características de los Requisitos

Las características deseables para una buena especificación de requisitos de software según lo indica la IEEE son las siguientes:

11.1. Conciso: Un requisito es conciso si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarlo en un futuro.

11.2. No ambiguo: Se considera que un requisito es ambiguo cuando puede ser interpretado de formas diversas por diferentes personas. Así, el estándar IEEE 830 establece que un requisito es No-ambiguo “si, y sólo si, puede estar sujeto a una única interpretación.”

El lenguaje usado en la definición del requisito no debe causar confusiones al lector.

11.3. Completo: Un requisito está completo si no necesita ampliar detalles en su redacción, es decir, si se proporciona la información suficiente para su comprensión.

11.4. Verificable: Un requisito es verificable cuando puede ser cuantificado de manera que permita hacer uso de los siguientes métodos de verificación: inspección, análisis, demostración o pruebas.

Se considera que un requisito es verificable “si existe un proceso acotado (en plazo y presupuesto) que permita determinar que el sistema construido satisface lo descrito en el propio requisito”.

Una forma de conseguir que los requisitos sean verificables es describirlos en suficiente detalle, o teniendo en cuenta que una de las premisas que debemos cumplir es que sean probados una vez implementados. Habrá requisitos que sólo

podrán ser verificados por simple inspección (existe una determinada funcionalidad o no) y otros que deberán ser probados (se deberán establecer una serie de pasos a seguir para comprobar que se cumplen.)

- 11.5. Consistente:** Un requisito es consistente si no es contradictorio con otro requisito (No existan contradicciones entre los requisitos).

Se considera que un requisito es consistente “si, y sólo si, no hay ningún otro requisito descrito dentro de un documento de especificación de requisitos que esté en conflicto con cualquier otro.”

- 11.6. No Redundante:** Un requisito es redundante si se encuentran requisitos expresando lo mismo.

Se considera que un requisito No es redundante “si, y sólo si, no hay ningún otro requisito que sea igual dentro de un documento de especificación de requisitos.”

- 11.7. Preciso:** Un requisito es preciso si hace uso de valores numéricos para precisar las características del sistema.

La precisión es aplicable, ante todo, a los requisitos no funcionales. Por ejemplo, no es útil decir “El tiempo de respuesta será más bien rápido”, sino “El tiempo de respuesta será menor que dos segundos”.

- 11.8. Clasificación:** No todos los requisitos son igual de importantes. Los requisitos pueden clasificarse por diversos criterios:

11.8.1. Importancia: Pueden ser esenciales, condicionales u opcionales. La clasificación de los requisitos por orden de importancia es una excelente práctica que permite establecer prioridades a la hora de abordar el desarrollo. Esta clasificación por el atributo “importancia” es necesaria desde un punto de vista práctico, dado que la construcción de sistemas se hace con un presupuesto limitado y en un plazo limitado.

11.8.2. Estabilidad: Cambios que pueden afectar al requisito. En el entorno de gestión, las necesidades del mercado imponen cambios continuos sobre los sistemas software, que se traducen, a su vez, en cambios en los requisitos.

11.8.3. Necesario: Un requisito es necesario si su omisión provoca una deficiencia en el sistema a construir, y además su capacidad, características físicas o factor de calidad no pueden ser reemplazados por otras capacidades del producto o del proceso.

Lo ideal es el establecimiento de prioridades, de modo que la implementación de un requisito de menor prioridad no emplee excesivos recursos.

11.9. Modificable: Se considera que un requisito es modificable “si su estructura es tal que permite realizar cambios sobre el requisito, manteniendo la estructura inicial del conjunto y continua siendo conciso, No ambiguo, completo y consistente. Los cambios son fáciles de introducir.

11.10. Trazable: Se considera que un requisito es trazable, si está claro el origen de cada requisito (quién o qué lo pide). Además, que todas las necesidades explícitas en el documento de requisitos estén contenidas en los casos de uso de negocio; es decir, que los requisitos se vean reflejados en al menos un caso de uso y que cada caso de uso corresponda a un requisito

12. Dificultades para definir los requisitos

A continuación se presentaran algunas de las principales dificultades para definir los requisitos:

- “Los Requisitos no son obvios y vienen de muchas fuentes.
- Son difíciles de expresar en palabras (el lenguaje nativo es ambiguo).
- Existen diferentes niveles de detalle.
- La cantidad de requisitos en un proyecto puede ser difícil de manejar.
- Nunca son iguales. Algunos son más difíciles, más riesgosos, más importantes o más estables que otros.
- Los requisitos están relacionados unos con otros, y a su vez se relacionan con otras partes del proceso.

- Cada requisito tiene propiedades únicas y abarcan áreas funcionales específicas.”¹⁸
- Un requisito puede cambiar a lo largo del ciclo de desarrollo.
- Son difíciles de cuantificar, ya que cada conjunto de requisitos es particular para cada proyecto.

13. Técnicas principales para obtener requisitos

“Para aplicar la ingeniería de requisitos en los proyectos de software esta es una tarea difícil ya que se requiere de un buen experto en el tema que tenga experiencia en esto y que tenga las habilidades de comunicación asertiva para realizar preguntas claves y ser muy buen analista. Es importante identificar a las personas involucradas en el proyecto para así saber cuáles son sus necesidades y que esperan del sistema ya que el entorno es muy cambiante y el sistema puede cambiar gracias a ese entorno. En este punto los analistas son libres de utilizar las mejores técnicas para la especificación de requisitos para así obtener buenos resultados cuando se están descubriendo las necesidades de los clientes, durante la historia y basados en la experiencia se dice que las técnicas de requisitos como las entrevistas, y talleres grupales dan muy buen resultado. Hoy en día hay nuevas técnicas como son los prototipos y casos de uso que son los que indican el funcionamiento del sistema. El analista cuando lo considere necesario empleará una combinación de estos buenos métodos para así realizar un buen descubrimiento de las necesidades de las personas implicadas para así crear un sistema que resuelva las necesidades del negocio.

Existen varias técnicas de requisitos, algunas creadas por diferentes empresas pensando en la adaptación de su proceso de desarrollo de software, pero las más usuales son:

13.1. Entrevistas

En las entrevistas se formulan preguntas claves y se seleccionan a las personas que tengan relación directa y crítica con el sistema, hoy en día este método es muy común y estas entrevistas pueden ser personales o grupales.

¹⁸ <http://www.mitecnologico.com/Main/ProcesosDeLaIngenieriaDeRequerimientos>

13.2. Talleres

Los talleres es un método muy efectivo, la mayoría de veces los requisitos no se descubren por medio de las entrevistas, y los talleres es una forma muy efectiva para descubrir las necesidades de los usuarios, estos talleres son facilitados por un analista de negocio, donde las personas involucradas participan grupalmente para descubrir los requisitos.

13.3. Forma de contrato

En lugar de una entrevista, se pueden llenar formularios o contratos indicando los requisitos. En sistemas muy complejos éstos pueden tener centenares de páginas.

13.4. Objetivos medibles

Los requisitos formulados por los usuarios se toman como objetivos generales, se debe realizar un análisis riguroso de esos objetivos para ir desglosando los requisitos en objetivos específicos.

13.5. Prototipos

Un prototipo es una interfaz donde se muestra el producto final terminados, es una pequeña muestra con funcionalidades limitadas, con estos prototipos se descubren nuevos requisitos porque los usuarios dan sus opiniones y mejoras al producto, los prototipos pueden ser diagramas y pantallas,

13.6. Casos de uso

Un caso de uso es una técnica donde describe los requisitos donde se grafica el sistema, y los actores involucrados con el sistema, los casos de uso son caja negra solo se muestra el funcionamiento del sistema, sus validaciones y restricciones. El objetivo de esta práctica es mejorar la comunicación entre los usuarios y los desarrolladores, mediante la prueba temprana de prototipos para minimizar cambios hacia el final del proyecto y reducir los costes finales.”¹⁹

¹⁹ Ingeniería De Requisitos Ingeniería De Software, <http://www.monografias.com/trabajos6/resof/resof.shtml>, 10/01/1 – 17:00

14. Características de un analista de requisitos efectivo

A continuación se describen algunas características que debería tener un buen analista de requisitos:

- “Cualificarse en conocimiento experto sobre ingeniería de requisitos y prácticas sobre requisitos. Tácticas para adquirir conocimiento pueden ser la asistencia a conferencias de ingeniería de requisitos.”²⁰
- Ser un buen comunicador, escritor y aprender a escuchar a los demás, tener don de gente. Estas tácticas permitirán al analista entender de una mejor manera las necesidades de los clientes y de los usuarios.
- Cualificarse en habilidades para la negociación, conciliación y mediación. De esta manera se facilitan los consensos entre los diferentes puntos de vista de todos los integrantes del equipo de trabajo, incluyendo los clientes y/o usuarios.
- Ser persistente y perseverante para aumentar las posibilidades adquirir y entender las verdaderas necesidades de los usuarios. De esta manera se reduce el riesgo que los usuarios rechacen el sistema o software.
- Ser proactivo invitando a que clientes, personal del proyecto, y gerente del proyecto a ayudar en la definición de los requisitos reales del sistema.
- Aprender y aplicar prácticas efectivas para el éxito del proyecto. El analista debe ser una persona en continuo aprendizaje de nuevas experiencias de estudio de tal manera que las cosas se hagan de una manera diferente, pero para bien.
- Ser claro en sus posiciones, y serio con sus relaciones y acciones.
- Desarrollar la habilidad para la estimación de tiempo y recursos para finalizar el trabajo técnico en un proyecto. El analista puede trabajar con el gerente del

²⁰ Tomado de www.monografias.com/.../traduccion-4-capitulo-del-libro-requeriments-handbook.pdf, 27/04/11 – 18:00

proyecto y el equipo desarrollador para realizar estimaciones, basándose en su previa experiencia.

- Mantenerse concentrado y enfocarse en la misión principal del proyecto, evitar tratar de caer en mal gastar tiempo en requisitos de menor importancia. Se recomienda por ello priorizar los requisitos de manera apropiada
- Estar al día de nueva tecnología y de cómo puede ser aplicada para resolver las necesidades de los clientes.
- Apuntar a metas alcanzables, persiguiendo siempre la idea principal del proyecto. El analista de sistemas debe lograr estas metas apoyándose por medio del documento de plan de proceso como ayuda.
- Hacer la diferencia en el trabajo profesional.
- Participar en la creación del proceso de riesgos, el analista puede ofrecer otro punto de vista ya que todo requisito está amenazado por un riesgo

ESTUDIO DE LOS PROCESOS ACTUALES DE PRUEBAS EN REQUISITOS EN LAS EMPRESAS.

a. Análisis del proceso

Existen muchos métodos útiles que se podrían implementar para determinar el estado de las pruebas en requisitos en las empresas de calidad. En esta investigación se utilizarán dos métodos: análisis de escenarios y pruebas de campo. Existen diferentes razones por las cuales se utilizaron estos métodos; En primer lugar, uno de los métodos más confiables para determinar cuál es el comportamiento de una población, es hacer un extenso análisis estadístico de casos representativos; sin embargo, por motivos de recursos de tiempo, personal y dinero; llevar a cabo una investigación a través de este método sería un objetivo difícil de alcanzar.

Por tanto, es necesario plantearse hipótesis de cómo podría ser el estado de las pruebas en requisitos, y en general, la ingeniería de los mismos en las empresas sujetas a investigación. Para esto, es necesario formular las posibles condiciones en que se pueden encontrar las mismas y establecer cuáles son las posibles características que se poseen los procesos de pruebas de requisitos en estas empresas.

Se realizó un estudio de dos empresas en el mercado nacional de pruebas de requisitos, por motivo de confidencialidad se omitirá el nombre de ellas, estas empresas ofrecen el servicio de pruebas de requisitos, pero su proceso no se encuentra suficientemente maduro, ya que las pruebas realizadas no son muy efectivas, o quizás no realizan dichas pruebas de manera exhaustiva de la cual se explicara a continuación.

El proceso que estas empresas utilizan para abordar la validación de requisitos, es realizar en primera instancia, una reunión de contextualización con el cliente, luego de que el analista de pruebas tenga claro el alcance, el cliente hace entrega de los requisitos por escrito, y el analista de pruebas procede a su lectura y análisis; luego, por medio de una lista de chequeo, evalúan que cada requisito cumpla con los parámetros exigidos en dicha lista de chequeo; unos de los aspectos más importantes de esta lista de chequeo son:

REQUISITOS
¿El requisito está escrito en un lenguaje personalizado utilizando la terminología del cliente?
¿Los requisitos funcionales transmiten suficiente conocimiento que permita empezar o continuar con las actividades de diseño?
¿El nivel de seguridad está especificado?
¿Está especificada la fiabilidad incluidas las consecuencias de falla de software, información vital protegida contra el fracaso, la detección de errores, y la recuperación?
¿Las compensaciones son aceptables entre los atributos de competencia especificados, por ejemplo, entre la robustez y la corrección?
¿Existen niveles de éxito o fracaso que permitan evaluar los requisitos no funcionales?
¿El mantenimiento del sistema está especificado, incluyendo la capacidad de responder a los cambios en el entorno operativo, las interfaces con otros programas, precisión, rendimiento y capacidades adicionales de predecir?
¿Todos los objetivos de desempeño se encuentran propiamente especificados (disponibilidad, tiempos de respuesta, usuarios concurrentes, uso de canal)?

En caso de encontrarse un error, el analista de pruebas lo reporta en el bugtacker, luego cuando toda la ejecución de la prueba de requisitos este al 100% finalizada, se realiza un segundo ciclo de pruebas en donde simplemente se verifica la inconsistencias encontradas en el ciclo 1 de la prueba de requisitos. Una vez, todo los requisitos estén en el estado “aprobado” se realiza una carta de aceptación donde se indica que los requisitos pasaron por un proceso de calidad y fueron aceptados de manera correcta.

b. Metodologías y tecnologías actuales

Las metodologías en las cuales se basan estas empresas para ejecutar su proceso son:

- ISTQB: La norma internacional ISTQB, es en la actualidad la norma superior que certifica la calidad de los profesionales que intervienen en el testing de alto nivel. La certificación ISTQB plantea un esquema de calidad internacional para el desarrollo de software.
- ISO9001: Es un conjunto de normas sobre la calidad y la gestiones. La Norma ISO 9001 ha sido elaborada por el Comité Técnico ISO/TC176 de ISO Organización Internacional para la Estandarización y especifica los requisitos

para un buen sistema de gestión de la calidad que pueden utilizarse para su aplicación interna por las organizaciones, para certificación o con fines contractuales.

- CMMI : (Modelo de Madurez de Capacidad Integrado) pertenece a la familia de modelos desarrollados por el SEI (Software Engineering Institute) para evaluar las capacidades de las organizaciones de ingeniería de sistemas, ingeniería de software, además del desarrollo integrado del producto y del proceso. es un modelo descriptivo que detalla los atributos esenciales que deberían caracterizar a una organización en un determinado nivel de maduración.

Los artefactos que utilizan son:

- Para la ejecución utilizan una lista de chequeo en formato de Excel.
- Para los reportes utilizan una plantilla de Excel donde reportan el requisito.
- Una carta de aceptación que está elaborada en Excel.
- Un repositorio que esta creado en un servidor donde esta toda la información del proyecto y la pruebas realizadas.

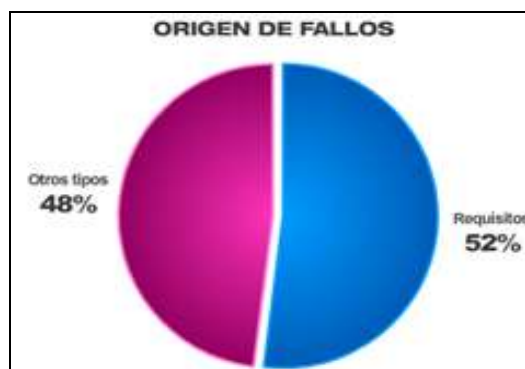
c. Problemas o posibles mejoras

Las pruebas de requisitos, pueden considerarse como uno de los procesos más complejos e importantes que hay en el ciclo de vida del software, dado que una buena elicitación permite comenzar el desarrollo con buenos requisitos en estructura y diseño, además está afectando en términos de reproceso costos y tiempo, que es un factor bastante considerable en un proyecto de desarrollo de software. Según el análisis realizado puede determinarse a simple vista, que el estado de madurez del proceso puede fallar en algunos aspectos, la listas de chequeo son muy generales y no abordan toda la estructura del requisitos, falta una trazabilidad requisitos vs casos de uso, falta definir y dejar por escrito, una priorización y riesgos en los requisitos, y por último está generando reprocesos en la documentación. Adicionalmente, no se evidencia una estructura definida del proceso a seguir porque el proceso no está definido formalmente, solo está definido una lista de chequeo.

Algunas empresas de calidad piensan que con un check list están aprobando la calidad de los requisitos ya que no tienen una estructura formalmente definida como la tiene las pruebas funcionales, lo que motiva a definir un proceso estructurado con miras a lograr la madurez del mismo a futuro, en donde su complejidad de ejecución sea baja y el tiempo de operación pueda disminuirse notablemente.

1. Situación Actual

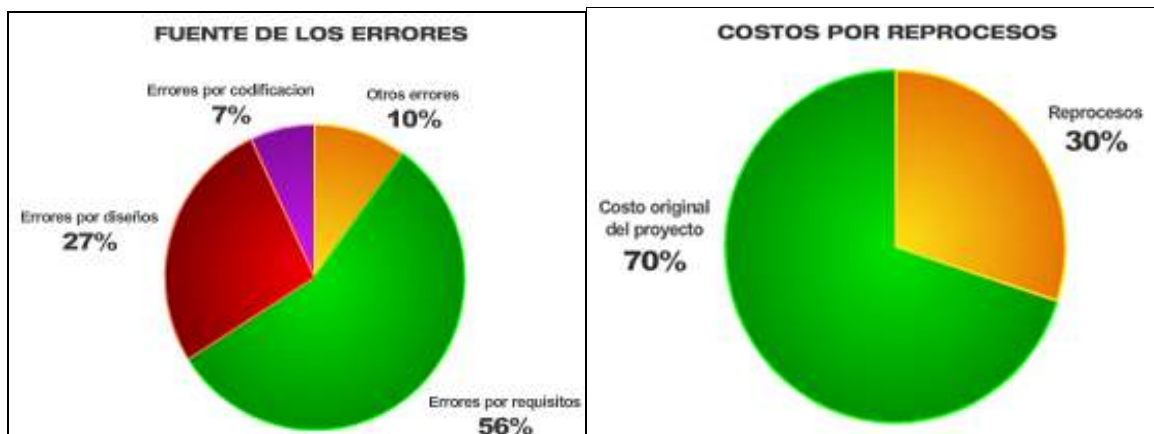
“Estudios realizados demuestran que sólo el 29% de los proyectos finalizan con éxito. Entendiendo por proyectos exitosos, aquellos que terminan en el tiempo estimado, presupuesto establecido y funcionalidad requerida. Lo anterior, evidencia un alto número de proyectos que fallan al tratar de alcanzar sus objetivos.





21

A partir del estudio de las principales causas de fallas en los procesos, se encuentra que el 52% se atribuyen a deficiencias en el proceso de ingeniería de requisitos como son: una mala administración de los requisitos, trabajar con stakeholders inapropiados, poca participación de los usuarios, entre otros. Todo lo anterior genera un alto nivel de reproceso de actividades que deben realizarse para corregir dichas fallas. Una valoración de los costos por reproceso muestra que estos pueden llegar a representar el 30% del total del costo del proyecto.



²¹ Estadísticas tomadas por la empresa Capital Empresarial, www.capitalempresarial.com.co, 16/04/11 – 13:00



22

Estos costos pueden llegar a aumentar dependiendo de la etapa del proyecto en la cual sean identificados. La detección temprana de estas fallas ayuda a disminuir considerablemente el impacto en todos los aspectos del proyecto, incluido el costo de corrección. En contraposición, la detección tardía de una de estas fallas puede llegar a aumentar los costos de corrección hasta 1000 veces que si se hubiese hecho al inicio del proyecto, esto sin considerar los impactos de imagen y perjuicios comerciales que pudiese llegar a tener un error de alto impacto en un ambiente productivo.”²³

2. Las empresas de servicios en EEUU y Europa

A continuación, se realizó un estudio del mercado de pruebas en EEUU y Europa, donde se quiere mostrar cómo están las empresas en ahorro de recursos, tiempos y los costos de encontrar errores en etapas de prueba, y estimaciones en costos de la tendencia de consumos de pruebas en los próximos años.

Computer World Federal es una empresa que estudió el costo en los EE.UU., sobre los desarrollos de software, que son aproximadamente de 59,5 mil millones dólares por año. El estudio mostró además, que si a estos proyectos se les realiza pruebas de software, podría reducir este costo aproximadamente en 22,5 mil millones dólares por año (más de un tercio). Los estudios arrojan que cuatro de cada cinco proyectos de Tecnologías de Información (TI) no

²² Estadísticas tomadas por la empresa Capital Empresarial, www.capitalempresarial.com.co, 16/04/11 – 13:00

²³ Tomado de Ingeniería de Requisitos, www.capitalempresarial.com.co, 16/04/11 – 13:00

contaron con pruebas y de estos proyectos el 90% se sometieron a sobrecostos, con todo este estudio se puede demostrar que las pruebas de software no son una moda, sino que tienen una alta necesidad en los proyectos de software.

Adicionalmente, en este estudio se centraron en los siguientes sectores: la banca, los seguros, las empresas de biotecnología, ciencias de vida (Farmacéuticos). En Nueva York solamente, hay más de 175 empresas independientes de servicios financieros con ingresos anuales de más de \$ 250 mil millones de dólares y en Nueva Jersey los Farmacéuticos dominan el mercado.

Ambas industrias están altamente reguladas, competitivas y requieren de los más altos estándares de calidad tanto a nivel interno, como en los productos y servicios que ofrecen al público. Ellos no quieren perder tiempo o arriesgar su dinero en busca de soluciones de negocio o conocimientos cuestionables.

2.1. Segmentación del mercado

El aseguramiento de calidad (QA) ha evolucionado más allá de la aceptación en el mercado, ha demostrado la necesidad de estar dentro del proceso de ingeniería de software, estando presente en cada una de las etapas del ciclo de vida de desarrollo de software, empezando por los requisitos, hasta la implementación, y con la inclusión de este proceso, han demostrado la calidad, los bajos costos y la reducción en los tiempos del proyecto.

El crecimiento de los ingresos y penetración en el mercado de los fabricantes líderes de software relacionados con las pruebas, apoya esta teoría. Una de las empresas líderes en el mercado es Mercury Interactive que ha experimentado un crecimiento anual del 45% en sus ingresos, la compañía está en 25 países, tiene más de 1.600 empleados y su participación en el mercado es de más del 50% con más de 30.000 clientes.

Las políticas operacionales, los procesos y procedimientos no se desarrollan adecuadamente, lo que limita el rendimiento de las empresas para lograr la inversión esperada.

Las organizaciones más exitosas han invertido adecuadamente en control de calidad e infraestructura de pruebas donde se han dado cuenta de los beneficios financieros y de negocios que trae adaptar esta metodología en las compañías.

A continuación se mencionarán algunos registros de incidentes presentados en EEUU:

- Los Bug's encontrados en EE.UU le cuestan a la nación más de \$ 60 mil millones de dólares al año, especialistas en pruebas indican que este costo se puede reducir en más de un tercio.
- El enfoque a corto plazo en los gastos financieros se aprieta como la confianza empresarial para TI continúa aumentando. A pesar de las restricciones presupuestarias, la infraestructura de TI todavía tendrá que cumplir con las iniciativas empresariales de la organización. La subcontratación y los proveedores de confianza tendrán un mayor control ya que el gasto de capital se reduce a favor de los presupuestos de funcionamiento como son las pruebas.
- La mayoría de las organizaciones de TI financieros, se están embarcando en iniciativas de integración a gran escala de aplicaciones para generar la innovación empresarial y para mantenerse al día con las migraciones de tecnología como Microsoft Windows XP y los temas de la industria STP (Straight Through Processing). Los especialistas en estas áreas garantizan unos productos de software con aseguramiento de la calidad.
- El mercado de las pruebas de software estaba valorada en 1,11 mil millones dólares en 2001, las pruebas están en constante aumento formado por cerca del 50% de esta en \$ 610 millones de dólares. El mercado de las pruebas aumentó a \$ 740 millones en 2002, para el 2005 el crecimiento fue del 21,3%, para el 2006 aumento en un 80% y en el 2010 el mercado de las pruebas de software está en 10 mil millones de dólares en EEUU.

Esto dejó que los servicios no distribuidos y los productos en torno de las pruebas de software compartían una cuota del 20% de \$ 1000 millones en los años 2009, de los cuales Norteamérica consumía aproximadamente dos tercios, y Europa Occidental alrededor de un cuarto.

Según estudios de Infosys, el señor Jefe Ejecutivo y Gerente Kris Gopalakrishnan, estimo que las pruebas de software pueden alcanzar hasta unos 13 millones en el 2011 en ventas, de esta estimación la mitad del trabajo en pruebas se encuentra en la india, todo esto fue discutido en la conferencia que se realizó en Bangalore y organizado por step-in. El objetivo de la conferencia era discutir sobre las diversas oportunidades que tiene el mercado de pruebas de software y el futuro de la india en este mercado.

2.2. Empresas en EEUU y Europa que realizan Pruebas de Requisitos

Algunas empresas que realizan Pruebas de Requisitos en EEUU y Europa son:

- **Test Labs** - (<http://www.offshoretestingervices.com/>)



La empresa fue fundada en 1997. En 12 años de existencia han logrado el reconocimiento de ser uno de los primeros en la industria del software en la India.

TestLabs es una compañía con filiales en Europa que proporciona desarrollo de software offshore, multimedia y servicios de pruebas.

Durante la última década, han apoyado a más de 50 empresas contando con excelentes resultados. La compañía cuenta con ingenieros expertos en pruebas de software. TestLab también cuenta con 35 ingenieros de automatización de pruebas, estos ingenieros son expertos en el manejo de las diferentes herramientas automatizadas, como QTP, TestComplete, LoadRunner, QFTTest, WAPT, OpenSTA y muchos más.

- **A1QA** - (<http://www.a1qa.com/>)



Compañía líder en el mercado de EEUU en la prestación de servicios de todo el ciclo de calidad y aplicaciones de pruebas.

A1QA posee varios laboratorios de pruebas en Europa, lo que les permite mantener la calidad de sus procesos y metodologías para reducir costos operativos, para los clientes significa que pueden recibir todas las prestaciones del servicio a bajos costos.

La compañía más allá de los conocimientos técnicos, se centra en sacar las mejores cualidades y capacidades. En marzo de 2008 abrió la Academia de control de calidad, y un centro de intercambio de conocimientos, su objetivo es dotar a los empleados y los equipos de control de calidad con el conocimiento, con el fin de promover el desarrollo de cada uno de los individuos de la misma.

- **uTest** - (<http://www.utest.com/>)



Esta compañía fue fundada en el 2008 en EEUU, hoy en día una de las compañías líder en el mercado de pruebas de software,

uTest No sólo ofrece servicios de pruebas de software, también los siguientes productos:

- Integración con los sistemas de seguimiento de errores, incluyendo Jira, Rally y Bugzilla.
- Se ha añadido un sistema con características que ayudaron a los clientes a construir mejor su propio equipo virtual de control de calidad.

3. Las empresas de servicios en Colombia

3.1. La situación actual de las empresas de desarrollo de software

Las pequeñas empresas forman parte del porcentaje más alto de empresas que mueven el motor de la economía nacional. Esto se debe, a que junto con la mediana empresa, constituyen el 90% del parque nacional empresarial. La población de las mismas es cinco veces mayor que la población de medianas empresas. Por estas razones, constituyen una oportunidad de investigación muy importante cuando se trata de establecer elementos que colaboren al mejoramiento de la realidad económica nacional.

Estudios realizados alrededor de las PyMEs han mostrado el verdadero panorama que enfrentan las pequeñas empresas. Entre los diferentes aspectos que rodean al mismo, se puede destacar que: El 50% de las PyMEs ha tenido reducciones en inversión, el 43% no han solicitado créditos en los últimos años, enfrentan cerca de sesenta modalidades de tributación diferente, el 87% de PyMEs no exportan y el 50% afirma que en los últimos dos años sus utilidades se han reducido.

En la situación actual de las empresas de calidad de software de Colombia, se puede identificar que en el mercado se encuentra mano de obra calificada, y una gran oferta de ingenieros, sin embargo, hay una muy baja participación de certificaciones internacionales por parte de los mismos, cabe destacar que las empresas están emprendiendo un proceso para la aplicación de programas de calidad.

Esto contribuye a que las empresas sean más competitivas y se superen barreras en nuevos mercados, ya sean nacionales como internacionales. En este aspecto, sólo un bajo porcentaje de empresas cuentan en la actualidad con la certificación en ISO 9000, incluyendo a grandes, medianas y pequeñas empresas. Estos resultados son muy alarmantes comparados con otros sectores industriales, los cuales han sido forzado a aplicar modelos de Calidad que eleven el nivel del sector. Las Empresas Colombianas están en desventaja frente a niveles de competitividad basados en la aplicación de programas de Calidad. Estados Unidos cuenta con aproximadamente 1700 empresas

que aplican programas de CMM y CMMI. En Colombia no llegan a 10 las empresas que implementan estos modelos.

3.2. Las Pruebas de Requisitos

En el campo de las pequeñas empresas existen muchos interrogantes. No se conocen estadísticas concretas de cómo se llevan los procesos relacionados con el desarrollo de software, ni de sus procesos (si son o no efectivos) las pruebas de requisitos.

Analizar a las empresas sus procesos, su calidad, y cómo mejorar dichos procesos, en las empresa desarrolladora de software, es un campo de investigación muy amplio.

Un estudio de proyectos de software revelo que solo el 12.7% de proyectos se desarrollan satisfactoriamente, mientras 76% de los proyectos se suspenden o fracasan, esto se debe a la baja calidad de los requisitos de software, y los principales problemas de los requisitos están asociados a: ambigüedad, concisión y contraposición. Asegurar la calidad de los requisitos en una tarea muy importante para poder terminar un proyecto de software con éxito. Se sabe que a medida que aumenta la etapa del proyecto y se encuentra un error, el costo será más alto, pues el costo de corregir un error en la última etapa está medido en 60 y 100 veces más, que el costoso de corregirlo en las primeras etapas.

En la siguiente figura se muestra que hay una etapa de pruebas de aceptación en las cuales se pretende probar el sistema y corregir sus errores. La corrección de estos errores en esta etapa de pruebas es mucho más costosa que si los errores se corrigieran en la etapa de requisitos o de análisis de sistema. Por tal motivo vemos la gran importancia de adelantar un proceso de pruebas en la etapa inicial que es la de requisitos, esto se debe utilizar como estrategia para disminuir errores graves o de alto impacto dentro del proyecto,.

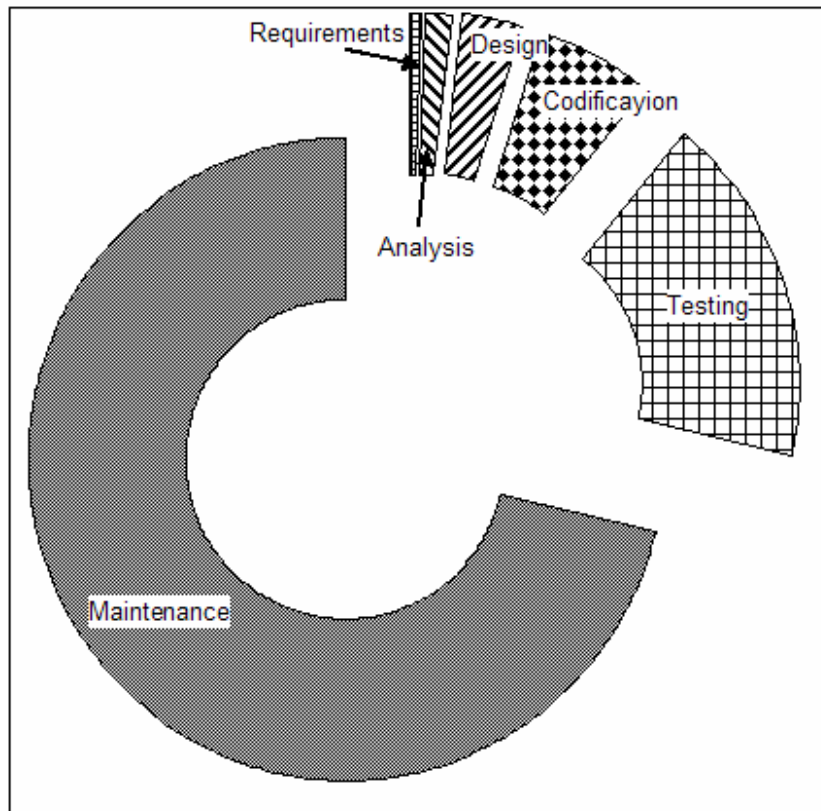


Fig13. Costo de corregir errores en las distintas fases del desarrollo.

3.3. El estado de las pruebas de requisitos en las empresas.

Una de las empresas de testing a nivel local, ofrece su servicio de pruebas de requisitos a una empresa Colombiana la cual tiene gran volumen de proveedores de software que son los que les desarrollan los productos a la compañía. Esta empresa local, realizó pruebas de requisitos a cada uno de los insumos entregados por dichos proveedores, en donde el resultado no fue muy favorable, pues el porcentaje de defectos en los requisitos era bastante alto. Esto nos pone a pensar que muchas de las compañías de nuestro país tienen aplicativos en producción, que le sirven a la comunidad en general y que convivimos con estos errores, ya que estas grandes compañías no realizan pruebas de requisitos, ni pruebas funcionales, ya que creen que realizar este tipo de labor la deben garantizar las empresas prestadoras de servicios como las casas desarrolladoras de software.

La problemática del país puede considerarse como un problema a nivel cultural porque se cree que las pruebas solo causan aumento de costos y tiempos. Las empresas necesitan dichas pruebas para disminuir tiempos y costos, ahí entran a jugar las empresas de calidad, demostrándole a estas compañías de Colombia que es necesario dentro de todo el ciclo de desarrollo tener un aseguramiento de la calidad en cada etapa del desarrollo. Con esto se podrá evidenciar la importancia de realizar pruebas en requisitos ya que al detectar un error al principio cuesta menos que detectarlo al final del ciclo de desarrollo y en Colombia son pocas las empresas que realizan pruebas de requisitos.

Adicionalmente, realizamos un estudio a 30 empresas colombianas por medio de unas visitas que se le realizaron donde la pregunta específica si realizaban pruebas de requisitos, estas empresas son (Bancolombia, Direct TV, Babaria, Tuya, Sufinanciamiento, Suramericana, Protección, Comcel, Tigo, Axede S.A, UNE, EPM, Davivienda, DAS, Comfama, Familia Sancela, Bolsa y Renta, Transito de Medellín, Banacol, TATA, EDATEL, Price wáter house coopers, Mi buñuelo, Ecopetrol, Postobon S.A, ISA, ISAGEN, Banco de Bogotá, AVON, ÉXITO S.A.)

Estas empresas en su mayoría no aplican en sus procesos de calidad pruebas de requisitos, solo 3 de 30 empresas demandantes de servicios aplican pruebas en requisitos, es decir que el 10% de este muestreo de empresas que se le realizó el estudio no tienen dentro de sus procesos de aseguramiento de calidad las pruebas de requisitos.

3.4. Empresas que realizan Pruebas de Requisitos

A continuación mencionaremos algunas de las empresas reconocidas en el medio Colombiano que realizan pruebas de requisitos:

- **Quality Vision Technologies S.A. (Q-Vision S.A.)** - (www.qvision.com.co)



Q-Vision es una compañía con trayectoria internacional que se especializa en la prestación de servicios en áreas de ingeniería de software, gestión de tecnología, aseguramiento y control de calidad de software (QA/QC), en organizaciones que utilizan y/o desarrollan software como soporte a la operación de su negocio.



- **Choucair Testing S.A – Colombia –** (www.choucairtesting.com)



Choucair Testing S.A. es una empresa colombiana con Proyección Internacional creada en el año 1999 y es especializada en prestar el servicio de Pruebas de Software en las diferentes etapas del ciclo de desarrollo de software

- **Delonia Software** – (<http://delonia.com/DeloniaWeb>)



Gestión de Requisitos

Es una compañía que considera que la Gestión de Requisitos es el eje fundamental de todo proyecto de ingeniería, ya sea en construcción, aeronáutica o informática. Tanto desde el punto de vista de la construcción del software como desde el aseguramiento de la Calidad y cumplimiento de Procesos, los requisitos deben ser gestionados correctamente dentro de todos los proyectos.

Ya sea mediante el seguimiento de procesos o metodologías o por ejecución de tareas rutinarias de gestión y control, en Delonia se adaptan los mecanismos de aseguramiento de calidad a las circunstancias de cada proyecto con los siguientes objetivos:

- Documentar debidamente los requisitos funcionales, técnicos y operativos del proyecto
- Definir mecanismos de validación (pruebas funcionales, stress, validación de usuarios, etc).
- Maximizar la visibilidad del proyecto mediante entregas continuas, permitiendo validación progresiva de requisitos:
 - Presentaciones
 - Video Simulaciones
 - Maquetas
 - Prototipos
 - Releases Alpha, Beta, RC
- Permitir la evolución y cambio de requisitos a lo largo de la vida del proyecto, minimizando el impacto en tiempo y coste del mismo.
- Asociar y gestionar correctamente defectos y peticiones de cambio, evaluando esfuerzos, costes e impactos en proyecto.

El objetivo final del aseguramiento de la calidad, en lo referente a la gestión de requisitos, es obtener el producto que desea el cliente con la máxima eficiencia en plazos y costes de proyecto.

- **Nexos Software S.A.S – Colombia** – (www.nexos-software.com.co)



Nexos Software S.A.S. nace el 31 de mayo de 2000 en la ciudad de Medellín Colombia. Desde el inicio la empresa fue orientada a la prestación de servicios de desarrollo de software a la medida en la modalidad de outsourcing, por ello su evolución ha sido paralela con el avance tecnológico y el incremento en diversidad de plataformas.

La metodología consta de las siguientes etapas:

- **Elicitación de requisitos:** esta es la etapa donde se adquiere el conocimiento del trabajo del cliente/usuario, se busca comprender sus necesidades y se detallan las restricciones. Como resultado de las acciones realizadas se tiene el conjunto de los requisitos de todas las partes involucradas. Existen algunas técnicas que utilizan para esta etapa como son Entrevistas, Técnica JAD de IBM y Casos de Uso definidos por UML.
- **Análisis de Requisitos:** En esta etapa estudian los requisitos extraídos en la etapa previa con el fin de poder detectar, entre otros, la presencia de áreas no especificadas, requisitos contradictorios y peticiones que aparecen como vagas e irrelevantes.
- **Especificación de Requisitos:** Partiendo de lo elaborado en la etapa anterior se hace una descripción del requerimiento.
- **Validación y Certificación de los Requisitos:** Esta etapa final se nutre de las anteriores y realiza la integración y validación final de lo obtenido, como resultado final, en el Documento de Requisitos.

- **Software Quality Assurance S.A – SQA** – (<http://www.sqasa.com/>)



SQA S.A. realiza en forma integral proceso de pruebas, con el objetivo de certificar productos de alta calidad, suministrando personal especializado, metodologías de pruebas, herramientas y coordinación de proyectos.

- Verificación de artefactos de requisitos y análisis: Revisiones formales de la documentación de requisitos y casos de uso

ESTUDIO DE LOS PROYECTOS Y TIPIFICACIÓN DE PRUEBAS EN REQUISITOS.

En una de las empresas en estudio, se logró acceder a sus métricas (Tabla1) donde básicamente están desarrollando varios proyectos de complejidad alta y media, a los cuales se le realizaron pruebas de requisitos. Cabe anotar, que esta empresa no desarrolla sino que sus necesidades las tercerizan con proveedores y cada uno se encarga de uno o varios desarrollos. La tabla 1 contiene la siguiente información:

- #: Numero del proyecto
- Nombre del pedido: Significa el proyecto que está desarrollando el proveedor
- Cluster: Seleccionar el cluster al cual pertenece el requerimiento
- Tipo de prueba: Tipo de prueba que se aplicó es decir, si es de requisitos, es son funcionales, si es performance.
- Horas reales invertidas en la certificación: Son las horas que se utilizaron para certificar el documento de especificación de requisitos.
- Número de requisitos: El número total de requisitos que tiene el proyecto.
- Número de casos de uso: Es el número de casos de uso que tiene el proyecto, que fueron sacados de los casos de uso.
- Número de reportes generados: Es el número de errores totales encontrados en la prueba.
- Errores de requisitos: Es el número de errores totales encontrados en la prueba de requisitos.
- Errores de diseño: son el total de incidentes encontrados durante la ejecución de las pruebas de requisitos para el diseño (casos de uso).
- Proveedor: Es la empresa externa encargada del desarrollo del proyecto y responsable de la especificación de los requisitos.
- Número de corridas: Es el número de ciclos de ejecución que se realizó para dejar el documento de especificación de requisitos con un aseguramiento de calidad.

#	Nombre Pedido	Cluster	Tipo Prueba	Horas Invertidas	# Requisitos	# Casos de Uso	Reportes Generados	Errores de Requisitos	Errores de Diseño	Proveedor	Numero de Correcciones
1	Titularizar Cartera FRECH	Cartera	REQUISITOS	5	3	0	4	4	0	Bancolombia	1
2	Vinculación a pensiones cor	PASIVOS	REQUISITOS	30	4	3	24	8	0	Bancolombia	4
3	Ajuste Aplicacion Sorteos T	TARJETAS	REQUISITOS	5	5	1	5	5	0	Bancolombia	1
4	Administración rangos de p	PASIVOS	REQUISITOS	22	9	5	26	26	0	Bancolombia	1
5	Fondo de Inversión Accione	CLIENTES	REQUISITOS	1	8	0	0	0	0	Bancolombia	1
6	CARTERA SOCIEDAD 2	Comercio, Filiales y Mi	REQUISITOS	24	7	0	19	19	0	GESTOR	1
7	CARTERA SOCIEDAD 2	Comercio, Filiales y Mi	REQUISITOS	13	7	0	10	0	10	GESTOR	1
8	Reversar cierre individual c	PASIVOS	REQUISITOS	37,5	2	0	22	12	0	Bancolombia	3
9	Mejoras del interactivo	CANALES	REQUISITOS	24,5	31	39	28	28	0	Bancolombia	2
10	Convivencia Canales ATM-P	CANALES	REQUISITOS	20,5	29	0	6	6	6	Bancolombia	1
11	Ajustes archivos por ciclos	CANALES	REQUISITOS	3	3	3	7	0	3	TATA	1
12	Dismunición de Fraudes	PASIVOS	REQUISITOS	9	6	1	4	2	2	Bankvision	1
13	Modificaciones MAT EMV A	TARJETAS	REQUISITOS	5	4	0	1	1	0	Bancolombia	1
14	Validación en Listas de Con	PASIVOS	REQUISITOS	2	2	0	13	9	0	Bancolombia	2
15	Certificados Retención en la	RIA, BANCA DE INVER	REQUISITOS	3	5	0	0	0	0	Heinsohn	1
16	Certificados Retención en la	RIA, BANCA DE INVER	REQUISITOS	6	5	0	0	0	0	Heinsohn	2
17	SLI - Manejo de decimales y	Comercio, Filiales y Mi	REQUISITOS	6	2	0	6	6	0	PersonalSoft	2
18	COFIM017 – Migración Bloq	INNOVA	REQUISITOS	1,5	0	0	0	0	0	Bancolombia	1
19	COMEM001 – Migración TRA	INNOVA	REQUISITOS	10	0	0	1	1	0	Bancolombia	2
20	Informe Airline Addenda M	TARJETAS	REQUISITOS	11	7	1	12	0	0	Bancolombia	2
21	Control automatico de tope	Canales	REQUISITOS	2	7	1	0	0	0	Bancolombia	1
23	Redeban Transmisión SFTP	TARJETAS	REQUISITOS	11	2	0	1	0	1	Bancolombia	2
24	COFIM011 - CDT panamá V1	INNOVA	REQUISITOS	1,5	0	0	0	0	0	Bancolombia	1
25	PMO12506_Modificación en respuesta de pagos de Ecopetrol	Comercio, Filiales y Mi	REQUISITOS	7	0	0	20	0	3	PersonalSoft	1
26	Venta de cartera improduc	Tesorería y Riesgos	REQUISITOS	13	19	0	1	1	0	Bancolombia	1
27	Venta de cartera improduc	Tesorería y Riesgos	REQUISITOS	4	19	0	1	0	1	Bancolombia	1
28	Convivencia Con Canales Su	Canales	REQUISITOS	57	2	0	2	2	0	Bancolombia	2
29	Cuentas Duplicadas	Canales	REQUISITOS	3	0	0	0	0	0	Bancolombia	1
30	Planilla Asistida Op no Bancarios	Canales	REQUISITOS	5	14	4	0	0	0	Bancolombia	1
31	Pago electrónico empleado	RIA, BANCA DE INVER	REQUISITOS	4	11	0	5	5	0	Bancolombia	1
32	OFAC-FEDLINK-SWIFT	Comercio, Filiales y Mi	REQUISITOS	5,5	5	0	8	8	0	Bancolombia	2
33	Reportes legales	Comercio, Filiales y Mi	REQUISITOS	5	2	0	6	6	0	Bancolombia	2
34	Informes Medios Magnético	Pasivos	REQUISITOS	4	2	0	6	6	0	Bancolombia	2
35	Cuentas Repetidas PGC	Canales	REQUISITOS	12,5	9	0	7	7	0	Bancolombia	2
36	Cambio cobertura pólizas	Pasivos	REQUISITOS	17	13	10	38	38	0	Bancolombia	1
37	PMO11422 ERP Caja Dolares	Comercio, Filiales y Mi	REQUISITOS	8	4	4	0	0	0	Bancolombia	1
38	PMO11422 ERP Caja Dolares	Comercio, Filiales y Mi	REQUISITOS	14	0	4	0	0	0	HP	1
39	Corrección de Cuentas Repetidas	Canales	REQUISITOS	1,5	6	8	0	0	0	Bancolombia	1
40	Corrección de Cuentas Repetidas	Canales	REQUISITOS	1,5	6	8	0	0	0	Bancolombia	1
41	Ajustes archivos por ciclos	Canales	REQUISITOS	5	3	1	2	2	0	Bancolombia	1
42	COFIM019 - Instrucciones Es	INNOVA	REQUISITOS	1	0	0	0	0	0	Bancolombia	1
43	COFIM020 - Migración Garar	INNOVA	REQUISITOS	2	0	0	1	1	0	Bancolombia	1
44	COFIM028 - Instrucciones Es	INNOVA	REQUISITOS	1,5	0	0	1	1	0	Bancolombia	1
45	COFIM041 – Migración Archi	INNOVA	REQUISITOS	1,5	0	0	0	0	0	Bancolombia	1
46	COFIM028 - Migración Instru	INNOVA	REQUISITOS	2,5	0	0	2	2	0	Bancolombia	1
47	COFIM012 instrucciones esp	INNOVA	REQUISITOS	1	0	0	2	2	0	Bancolombia	1
48	COFIM008 – Líneas de crédito	INNOVA	REQUISITOS	2,5	0	0	1	1	0	Bancolombia	1
49	COFIM013 - Garantías Panar	INNOVA	REQUISITOS	1	0	0	1	1	0	Bancolombia	1
50	COFIM013 - Garantías Panar	INNOVA	REQUISITOS	1,5	0	0	2	2	0	Bancolombia	1

Tabla 1: Métricas de un proyecto sobre pruebas en requisitos.

En la tabla 1 se muestran los proyectos atendidos durante n meses, su tipo de prueba, horas invertidas en cada proyecto, muestra el numero de requisitos y casos de uso que tiene cada proyecto, muestra el numero de errores encontrados durante un ciclo y el numero de ciclos aplicados a la prueba de requisitos.

En la tabla 1 se puede visualizar que en promedio se están haciendo dos corridas o ciclos de pruebas por proyecto, esto significa que siempre se están encontrando inconsistencias en el primer ciclo.

Las grandes empresas de Colombia manejan proyectos de alta complejidad, esto significa que se debe tener un muy buen proceso y administración en todas las etapas de desarrollo y se debe tener de carácter obligatorio dentro de la etapa inicial de requisitos un proceso de pruebas.

Se puede observar que esta tabla tiene una vigencia de 30 días, cada mes se sacan los indicadores de proyectos finalizados, lo que muestra 80 proyectos ejecutados en un mes y en todos se encontraron incidentes.

Las pruebas en requisitos deben ser ágiles y productivas, en donde asegure la calidad del sistema que se va a implementar, es decir, el proceso de pruebas en requisitos asegura la estructura y el diseño de los requisitos, pero no relación con el negocio.

En este punto y con el análisis de esta tabla es donde se le da importancia a las pruebas de requisitos, con esta tabla vemos que hay muchos errores de requisitos para un proyecto en esta compañía. La labor de elicitar y especificar los requisitos es difícil por el lenguaje tan extenso que manejamos acá en Colombia, y además que hay un porcentaje grande que por los requisitos mal especificados tienden los proyectos a cancelarse o en definitiva los llevan al fracaso, otra razón que nos pone a pensar con esta grafica se ha estudiado son los altos costos de encontrar un error en las etapas posteriores a la etapa de construcción, es decir si se encuentra un error cuando el sistema está en producción es de 60 a 100 veces más costoso que si se encuentra en la etapa inicial cuando se están especificando los requisitos.

ADAPTACIÓN DEL PROCESO

En el estudio realizado a algunas empresas que prestan el servicio pruebas, se evidencia que hoy en día dichas empresas realizan sus pruebas de requisitos a partir de la ejecución de unas listas de chequeo para validar la calidad de los requisitos, sin contar con un proceso maduro para ofrecer a los diferentes clientes que requieran de sus servicios.

Es por esto, que el objetivo para realizar este proyecto de grado, es la creación de un proceso para fortalecer las pruebas en requisitos, debido a que muchos de los problemas de la ingeniería de software provienen de la imprecisión en la especificación de los requisitos y a la suposición de que los requisitos proporcionados por el cliente son validos y correctos, lo cual es totalmente falso porque efectivamente son ellos los que conocen del negocio y los procesos, pero no son conocedores de las técnicas, procesos e implementación del software.

Un buen proceso de pruebas en requisitos nos facilitará en un futuro del desarrollo del software el cumplimiento de las expectativas o necesidades del cliente, esto evitará sobrecostos de implementación, generará en menor proporción reprocesos y aportará a la calidad del aplicativo desarrollado.

La detección de un error en etapas tempranas del proceso de software es menos costosa, por lo que las pruebas de requisitos toman mayor importancia, debido a que se efectúan o se aplican en una de las primeras etapas del ciclo de desarrollo del software y así en etapas posteriores se previene la generación de errores por malas especificaciones de los requisitos.

1. Generalidades

Las pruebas de requisitos están diseñadas para ser ejecutadas en paralelo a la etapa de Requisitos dentro del desarrollo de software y el objetivo es **apoyar** dicha etapa.

En sectores que se trabaja con las tecnologías de información, TI debe apoyar activamente la estrategia de la compañía del cliente.

Toda la compañía depende de las tecnologías de información tanto en los proyectos, procesos y planes de mejoramiento.

Los requisitos siempre deben reflejar correctamente las necesidades del cliente, y cumplir con ciertas características o propiedades para que la implementación de dichos requisitos, sean apropiados y acordes con la calidad y objetivos esperados del proyecto. Las pruebas de requisitos requieren más acompañamiento del usuario y de una participación más activa con respecto al conocimiento del negocio de los analistas de prueba. A diferencia de las pruebas de funcionales, donde el apoyo es más técnico que conceptual desde el punto de vista del negocio.

Para realiza unas buenas pruebas de requisitos se debe tener en cuenta lo siguiente:

- Debe existir artefactos de requisitos
- Se supone que los requisitos son correctos y completos desde el punto de vista de Negocio

1.1. Alta intervención del usuario

En el ciclo de vida de desarrollo de software, cuando se comienza la etapa de las pruebas de requisitos, se debe tener una alta intervención de usuario y poca intervención del analista de pruebas, ya que el usuario es la persona que tiene todo el conocimiento de las necesidades del proyecto, esta persona debe verificar que cada requisito contenga la especificación de lo requerido, y a medida que va avanzando las pruebas, la participación del usuario disminuye al mismo tiempo que la participación del analista de pruebas aumenta, pues la etapa de diseño se debe validar por el Analista de Pruebas ya que es quien tiene el conocimiento técnico.



Fig. 13. Ciclo de vida de desarrollo de software

1.2. Supuestos

Se parte del supuesto: ***“La idea del cliente es correcta”***

Como se mencionaba anteriormente, el cliente al inicio del ciclo de vida de desarrollo de software debe participar activamente en las pruebas, y el analista de pruebas es el que las dirige y realiza el acompañamiento al usuario, de tal manera que sea más fácil de definir los requisitos, encontrar inconsistencias y realizar observaciones pertinentes de mejora.

1.3. Alcance Producto Requisitos

Los requisitos nacen de una necesidad que detecta un mercado, un gobierno o una empresa, de ahí se realiza un análisis de negocio, y se establece un alcance para los requisitos, este alcance está considerado por el entorno como son cambios en el mercado, cambios de tecnología, mejoramiento de procesos, cambios estratégicos, solicitudes en espera que por costos o tiempo no se han logrado desarrollar, solicitudes de cambios de sistemas. Todo lo anterior viene del entorno y es de ahí de donde se debe realizar un análisis por un equipo de proyectos, este análisis se realiza teniendo en cuenta los objetivos de la empresa, y de ahí se definen los requisitos, luego pasan a una etapa de aprobación (si son aprobados, en espera, cambiados o rechazados) y luego, cuando se tienen todos los requisitos listo y aprobados por el clientes, se asignan al equipo de pruebas para que se enfoquen en esta etapa.

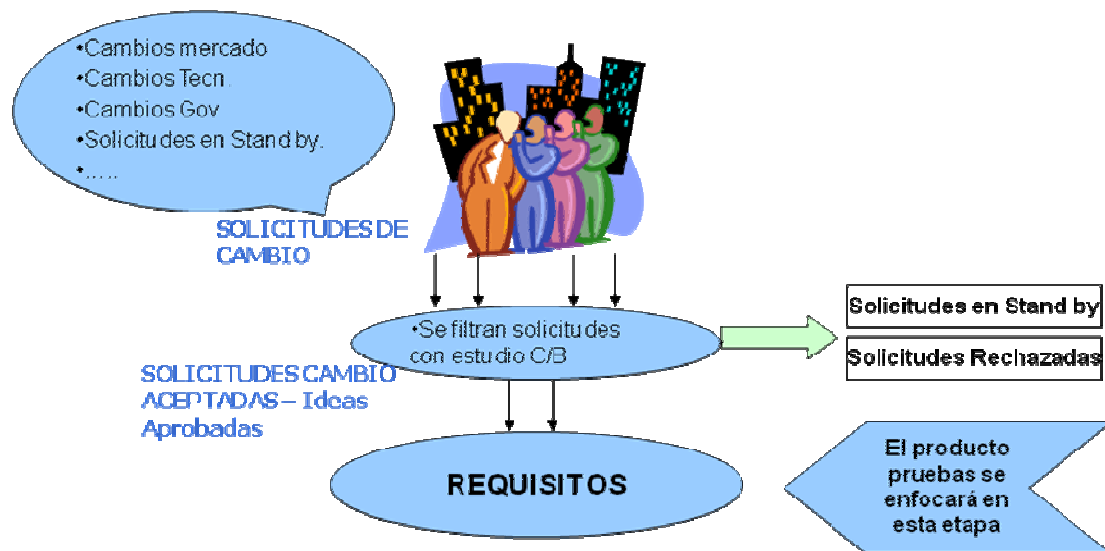


Fig. 14 tomada de Choucair Testing, www.choucairtesting.com.co

1.4. Qué se Prueba? – Procesos

El servicio está enfocado básicamente en verificar los requisitos (es decir, las solicitudes aceptadas), el Analista de Pruebas levantará las alertas correspondientes, para que el área de procesos verifique el impacto de la inclusión, modificación o retiro de los requisitos en sus procesos.

a. *Si el cliente tiene un modelo de desarrollo planteado:*

- Que los artefactos exigidos por el cliente existan
- Que los artefactos de los requisitos cumplan con las especificaciones de los artefactos del modelo de desarrollo planteado por el cliente.
- Verificar la existencia de los artefactos del modelado de los requisitos (la Modelación de los artefactos existan).

b. *Si el cliente NO tiene un modelo de diseño de desarrollo planteado:*

- No se puede hacer nada formal, queda a criterio de analista de pruebas

1.5. Que no incluyen las pruebas de requisitos.

El servicio de pruebas de requisitos no incluyen o se toman como supuestos lo siguientes ítems:

- No garantiza que la idea reflejada en el requisito sea la mejor forma de apoyar la estrategia o estrategia del negocio del cliente
- Estas pruebas no garantizan que el requisito sea la mejor solución desde el punto de vista del negocio
- No garantizan que se construya la mejor solución desde el punto de vista costo/beneficio

1.6. Requisitos para la prueba

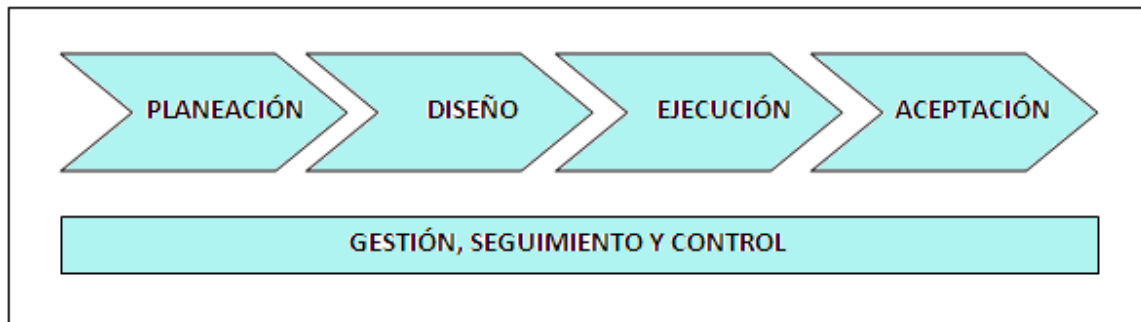
El servicio de pruebas de requisitos debe cumplir con unas reglas antes de empezar la prueba para poder garantizar un buen aseguramiento de la calidad en la etapa de requisitos, los cuales son los siguientes:

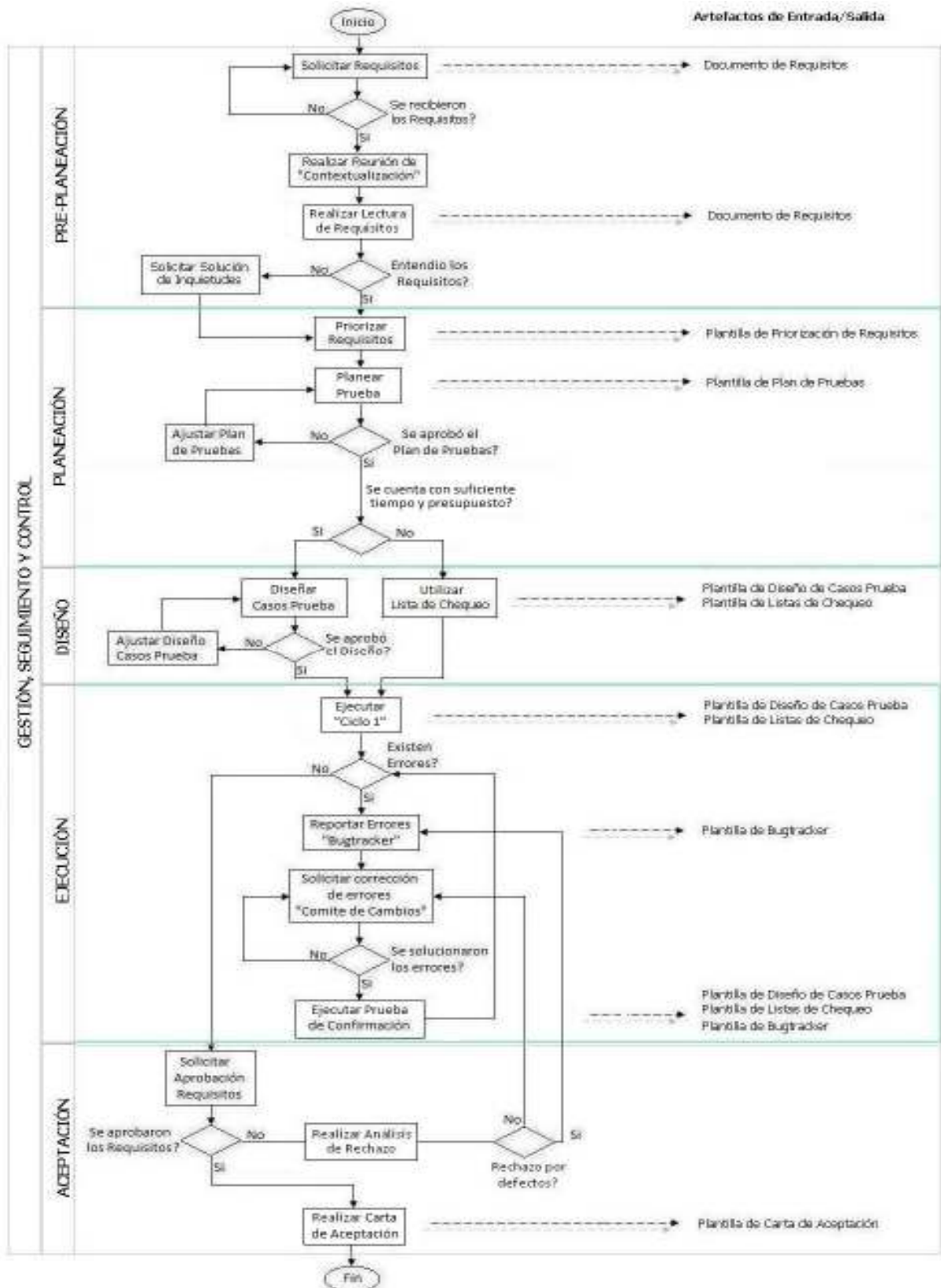
- Seguir detalladamente la metodología definida en este trabajo para las pruebas.
- Es ideal que el equipo del proyecto (distinto a pruebas) haya realizado una revisión inicial de los artefactos de requisitos, antes de la entrega para la ejecución de las pruebas.
- Para generar un mayor valor al producto, se hace necesario involucrarse en el negocio desde las estrategias y las tácticas.
- El analista de pruebas debe tener conocimiento del sector e idealmente sobre temas relacionados con el aplicativo que va a probar.
- Ideal conocer del mercado (productos utilizados en el medio) del cliente, para poder generar consideraciones de valor.
- Tener claras las reglas de negocio del cliente para el proceso al cual apoya el aplicativo.
- Tener claro cuáles artefactos de requisitos se deben exigir (artefactos exigidos por el cliente y por la metodología desarrollada).

- El analista de pruebas debe tener conceptos básicos de los temas de performance y seguridad.
- Nivelar al equipo que participa en el proyecto (clientes y/o tercero encargado del diseño), sobre las herramientas y metodología de pruebas que se van a utilizar.
- El cliente debe considerar un alto porcentaje de participación del usuario en la prueba.

2. Proceso propuesto de Pruebas de Requisitos

A continuación se mostrará el flujo del proceso propuesto para realizar una buena prueba en los requisitos especificados por los clientes y/o usuarios:





2.1. Pre-planeación

Para este tipo de pruebas se cuenta con un tiempo limitado lo que hace que unas de las tareas principales para el aseguramiento de los tiempos, es planificar de forma adecuada y certera dichas pruebas. Pero antes de comenzar con la etapa de la planificación de las pruebas, se debe solicitar todos los artefactos para el desarrollo del proceso de las pruebas, como los requisitos y casos de uso.

Dentro de la fase de ejecución se presentan las siguientes actividades:

2.1.1. Solicitar requisitos

Se debe solicitar todos los artefactos para el desarrollo del proceso de las pruebas, como los requisitos y casos de uso.

La verificación de la documentación tiene como objetivo la revisión de todos los artefactos entregados por el cliente que se encuentran definidos como entradas del proceso. Esta documentación es de carácter obligatorio y la ausencia de esta documentación hace que el proceso de pruebas no tenga la calidad esperada por el cliente.

- Los pasos para ejecutar esta actividad son los siguientes:
 1. Solicitar los requisitos al cliente y/o usuario.
 2. Recibir los requisitos por algún medio o dispositivo electrónico.
 3. Almacenar los requisitos.
- Artefactos de Entrada:
 - ✓ Correo electrónico
- Artefactos de Salida:
 - ✓ Documento de requisitos.
 - ✓ Descripción del sistema
 - ✓ Reglas de negocio.

Responsables:

- ✓ Analista de Pruebas.

- ✓ Clientes y/o usuarios.
- ✓ Director de proyectos

2.1.2. Realizar lectura de la documentación

Se debe seguir con la actividad de contextualización de la documentación, en este punto del proceso se realiza una lectura de todos los requisitos y si es necesario se solicita al cliente una capacitación de la documentación entregada.

Se debe analizar la documentación de entrada para realizar y definir varios puntos importantes dentro de esta etapa de planeación como su alcance, estrategia, priorización y estimación. El objetivo de este paso es poder complementar partes como el resumen de la documentación y una descripción del contenido recibido al iniciar el proyecto.

- Los pasos para ejecutar esta actividad son los siguientes:
 1. Leer los requisitos.
 2. Primer análisis del estado de los requisitos.
 3. Tomar apuntes de lo importante o dudas generadas durante la lectura.
- Artefactos de Entrada:
 - ✓ Documento de requisitos.
 - ✓ Descripción del sistema
 - ✓ Reglas de negocio.
- Artefactos de Salida:
 - ✓ Resúmenes de la documentación
 - ✓ Formulación de preguntas

Responsables:

- ✓ Analista de Pruebas.

2.2. Planeación

Una vez terminada la anterior actividad se pasa a una de las fases más importantes de todo el proceso de pruebas en requisitos, la etapa de planeación de pruebas, esta etapa es importante porque es el mapa del cual se guía y conoce desde donde se empieza y hasta donde se llega, como también cuánto tiempo se tardará durante el proceso de la prueba.

La planificación de las pruebas debe estar contemplada en estimaciones realistas, el objetivo principal de realizar una planificación es detallar todas las actividades a lo largo de las pruebas. Durante la etapa de planificación de las pruebas se deben tener todos los objetivos claros e identificados, se debe tener claro el tiempo estimado para el desarrollo del proyecto y así no perjudicar los tiempos de desarrollo, ni los tiempos que se estimen para esta prueba. Por eso es que se habla de unas pruebas de requisitos ágiles.

- Los pasos para ejecutar esta actividad son los siguientes:

2.2.1. Realizar priorización de los requisitos

Se debe priorizar los requisitos (se priorizan los procesos de negocio, y casos de uso según complejidad y riesgos asociados al negocio) en donde se ingresen en una plantilla y se evaluarán por medio de las siguientes variables:

- **Importancia para el Negocio:** Permite priorizar basado en el costo beneficio para el negocio. Para efectos de cálculos de prioridad de los requisitos este ítem tendrá un peso del 50%.

Se maneja la siguiente escala de calificación:

- 3 = Alta importancia para el negocio.
- 2 = Mediana importancia para el negocio.
- 1 = Baja importancia para el negocio.

- **Complejidad:** Permite priorizar basado en la dificultad de implementación del requisito. Para efectos de cálculos de prioridad de los requisitos este ítem tendrá un peso del 30%.

Se maneja la siguiente escala de calificación:

- 3 = Alta complejidad.
- 2 = Mediana complejidad.

➤ 1 = Baja complejidad.

- **Esfuerzo de Análisis:** Permite priorizar basado en la dificultad de análisis del requisito, teniendo en cuenta el tamaño (Cantidad de caracteres) y asociación con otros requisitos.

Para efectos de cálculos de prioridad de los requisitos este ítem tendrá un peso del 20%.

Se maneja la siguiente escala de calificación:

- 3 = Alto esfuerzo de Análisis.
- 2 = Mediano esfuerzo de Análisis.
- 1 = Bajo esfuerzo de Análisis

- **Resultado de Priorización:** Campo automático que presenta el resultado obtenido de la operación entre las calificaciones realizadas a cada ítem de priorización.

La operación realizada es la siguiente:

$$\{(\#Importancia \text{ para el Negocio}) * (\%Peso)\} + \{(\#Complejidad) * (\%Peso)\} + \{(\#Esfuerzo \text{ de Análisis}) * (\%Peso)\}$$

Luego se les dará el peso necesario para multiplicar las tres variables y da como resultado un peso sobre la prueba, este número significa la importación, prioridad y severidad del requisito, caso de uso, regla del negocio. Luego de tener todos los requisitos diligenciados en la plantilla con sus respectivos resultados, se organizan de mayor a menor, esto significa que entre mayor sea la priorización, el requisito será más importante para el sistema y esta priorización de requisitos sirve para asignarle una complejidad alta y tener mayor cuidado al verificar el requisito.

- Los pasos para ejecutar esta actividad son los siguientes:

1. Ingresar todos los requisitos en la plantilla.
2. Realizar la priorización de los requisitos, según los ítems requeridos en la plantilla.
3. Organizar descendentemente los requisitos, según el resultado de la priorización.

- **Artefactos de Entrada:**

- ✓ Documento de requisitos.
- ✓ Descripción del sistema
- ✓ Reglas de negocio.

✓ Plantilla priorización de requisitos.

• Artefactos de Salida:

✓ Plantilla priorización de requisitos.

Responsables:

✓ Analista de Pruebas.

Actividad	Rol	Entradas	Descripción de la actividad	Salidas
1. Definir el alcance	Analista de Pruebas	Inventario de requisito Reglas de negocio Casos de uso	Definición del alcance de las pruebas: Es donde se identifica que se prueba y que no se prueba, este alcance se saca de la siguiente documentación de entradas como son: descripción de sistema, catalogo de requisitos y especificación de requisitos. Otra manera que es muy efectiva para identificar el alcance son las reuniones que se realizan con el objetivo de definir el alcance.	El alcance de la prueba totalmente identificado en el artefacto "Plan de pruebas"
2. Estimación de tiempos	Analista de Pruebas	El alcance de la prueba Inventario de requisito Reglas de negocio Casos de uso	En esta actividad se estiman los tiempos que se dedicaran en todas las actividades que se realizaran en el transcurso del proyecto. Se detallan las actividades de acuerdo a lo definido en el alcance de la prueba, la estimación se realiza de la siguiente manera: <ul style="list-style-type: none">• Actividades a realizar• Numero de requisitos a probar• Numero de reglas de negocia a probar• Número de casos de uso a	Estimación de tiempo que se diligencia en el "Plan de pruebas"

			probar	
3. Definición del entorno de las pruebas	Analista de Pruebas	El alcance de la prueba Inventario de requisito Reglas de negocio Casos de uso	Definición del entorno de las pruebas, se debe garantizar que los requisitos estén totalmente terminados, esto quiere decir que ya se realizo toda la elicitación de requisitos, se especificaron los requisitos y se realizaron los casos de uso. Esto es de vital importación ya que no se puede desarrollar las pruebas si no se tiene este tipo de documentación terminada.	Entorno definido y diligenciado en el documento "Plan de pruebas"
4. Definir los recursos y para la gestión de las pruebas, utilización de plantillas y reporte de incidentes.	Analista de Pruebas	El alcance de la prueba Inventario de requisito Reglas de negocio Casos de uso	Se identifican cuantos recursos se requieren para la prueba, como también que herramientas se van a necesitar	recursos y herramientas
5. Definir riesgos de la prueba	Analista de Pruebas	El alcance de la prueba Inventario de requisito Reglas de negocio Casos de uso	Identificar los riesgos, donde se mencionan los principales riesgos de la prueba, su impacto y acciones a favor en caso de que se presenten esos riegos para su gestión y control. Los riesgos de una prueba son transversales esto quiere decir que hay que tenerlos en cuenta durante todo el desarrollo de la prueba.	riesgos para la prueba
6. definición de la estrategia	Analista de Pruebas	El alcance de la prueba Inventario de	Establecer las estrategias que se van a utilizar para realizar las pruebas de requisitos. Las estrategias de las pruebas deben	Estrategia de las pruebas

		requisito Reglas de negocio Casos de uso	estar alineadas con los objetivos del proyecto como son los plazos, costos y el aseguramiento de la calidad, se debe tener en cuenta el alcance y los tiempos estimados para estas pruebas.	
7. Enviar documento de plan de pruebas específico	Analista de Pruebas Director de proyectos	Plan de pruebas	Se debe enviar el plan de pruebas al director de proyectos para que le de la aprobación	Correo electrónico con dicha aprobación

Por último hay que tener en cuenta que el plan de pruebas puede sufrir modificaciones durante todo el ciclo de vida de las pruebas ya que puede suceder que durante las ejecución de las pruebas se detecten situaciones no consideradas para la pruebas, es importante que se maneje un historial para el documento, una línea base cada que se termine una etapa de la prueba, este historial contiene las fechas de creación, fechas de aprobación y descripción de evento.

2.3. Diseño

Un caso de prueba está conformado por un conjunto de entradas, condiciones de ejecución y los resultados esperados, para comprobar que cada requisito es correcto y la lógica del negocio.

- Los pasos para ejecutar esta actividad son los siguientes:

Actividad	Rol	Entradas	Descripción de la actividad	Salidas
1. Realizar la validación de los requisitos por el camino de diseño de casos de prueba	Analista de Pruebas	Plan de pruebas	Se deben especificar unos casos de prueba los cuales deben cubrir los requisitos documentados funcionales y no funcionales para realizar dicha validación.	Análisis de los casos de prueba

2. Verificar los escenarios del negocio	Analista de Pruebas	Plan de pruebas	Se verifican escenarios recorriendo de principio a fin la lógica de cada escenario. Para elaborar las pruebas , no solo necesitamos un camino de ejecución sino un resultado esperado de dicho camino, lo cual permite verificar si es satisfactorio o no satisfactorio el camino empleado, al verificar todas las posibles combinaciones es posible encontrar caminos de ejecución que no conduzcan a ninguna parte.	Escenarios identificados
3.Verificar escenarios no vistos en las especificaciones de requisitos	Analista de Pruebas	Documento "Plan de pruebas"	Identificar escenarios nuevos. Al verificar los escenarios es posible encontrar escenarios alternativos no documentados en los requisitos no funcionales.	Escenarios identificados
4. Verificar las reglas de negocio	Analista de Pruebas	Documento "Plan de pruebas"	Se deben verificar los casos de uso por medio de la estructura que debe tener cada caso de uso, como también el caso de uso describe los requisitos y las reglas de negocio.	Casos de uso identificados
5. Verificar la ambigüedad	Analista de Pruebas	Documento "Plan de pruebas"	Se debe verificar cada significado del requisito para así encontrar ambigüedades.	Diseño Casos de pruebas requisito x requisitos
6. Lista de chequeo para todos los requisito	Analista de Pruebas	Documento "Plan de pruebas"	Se debe verificar todos los requisitos de manera general por medio de una lista de chequeo estandarizada esto se hace para un mayor ahorro de tiempo	Lista de chequeo de requisitos general

7. Lista de chequeo para cada requisito	Analista de Pruebas	Documento "Plan de pruebas"	Se verifica cada uno de los requisitos por medio de una lista de chequeo estandarizada esto se hace para un mayor ahorro de tiempo	Lista de chequeo de requisitosxrequisitos
---	---------------------	-----------------------------	--	---

Las principales condiciones que se tendrán en cuenta en las listas de chequeo son las siguientes:

Generales
¿Los requisitos especifican una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo?
¿Los requisitos indican la posible solución del problema o necesidad (el cómo)? NOTA: Los requisitos deben expresar "el qué" de la necesidad.
¿Los requisitos dentro de su estructura contienen por lo menos la siguiente información: Tipo de requisito, Código del requisito, Nombre, Descripción del requisito, Autor o Responsable y Stakeholders impactados?
¿Cada requisito presenta un código de identificación claro y único?
¿Se identifica claramente el tipo de cada uno de los requisitos: Funcional, No funcional, De Información o Regla de negocio?
¿Cada requisito tiene una prioridad asociada?
¿Existe un glosario de términos que permita unificar los conceptos que se manejan en los requisitos definidos de forma clara y precisa?
¿Para cada uno de los requisitos se asocia una necesidad o caso de uso?
¿Se ha especificado el origen de cada requisito? (p.ej. quién lo pide, quien decidió incluirlo y porqué, etc.)
¿Se han considerado requisitos Funcionales y No funcionales?
¿Los requisitos están escritos en un lenguaje personalizado utilizando la terminología del cliente y/o Usuario?
¿Los Requisitos relacionan los aspectos que no se van a cubrir (lo que no se incluye) en el desarrollo?
¿Los requisitos se encuentran sin errores ortográficos y gramaticales?
Documento de Requisitos
¿El documento de Requisitos cuenta con por lo menos la siguiente información en el encabezado: Logo de la empresa, Título del documento, Nombre del proyecto y versión?
¿El documento de Requisitos cuenta con por lo menos la siguiente información en el Control del Documento "Registro de cambios": Versión, Motivo del cambio, Realizado por y Fecha?

¿El documento de Requisitos cuenta con por lo menos la siguiente información en el Listado de distribución: Área / Empresa, Nombres completo y Cargo?
¿El documento de requisito fue aceptado por todas las partes interesadas?
¿El documento de Requisitos cuenta con por lo menos la siguiente información referente a la Aprobación del documento: Firma Usuario Líder, Fecha Firma Usuario Líder, Firma Analista de Requisitos y Fecha Firma Analista de Requisitos?
Requisitos Empresariales
¿Los requisitos están orientados a los objetivos, misión y visión de la empresa?
¿Están identificados todos los Stakeholders (personas o áreas) que participaran o se impactaran con el proyecto?
¿Se ha identificado el “para qué” y/o el “valor agregado” para el negocio de los requisitos? (ahorro en costos, disminución en procesos, mejorar operación, reducir inventario, etc...)
¿Existe un glosario de términos de Negocio que permita aclarar los conceptos que se manejan en los requisitos?
¿Se han identificado los posibles riesgos del proyecto con su respectiva medición (Probabilidad e Impacto)?
¿Se analizó y documentó el posible impacto de la solución de las funcionalidades, módulos y/o aplicaciones a implementar en las diferentes áreas, sistemas o procesos de la compañía?
Reglas de Negocio
¿Las reglas de negocio describen las restricciones, leyes, límites y cualquier otra condición que puede restringir, condicionar o afectar la operación de un requisito funcional?
¿Se identifican las reglas de negocio dentro de los requisitos funcionales?
¿Se identifican las reglas de negocio en requisitos separados de los Funcionales?
Requisitos de Información
¿Se identifican los requisitos de información como datos necesarios para la operación del sistema?
¿Los requisitos de información describen cada uno de los datos que serán almacenados, presentados, convertidos y eliminados por la ejecución de un requisito funcional?
¿Los requisitos de Información indican los datos de entrada en el sistema incluyendo su fuente, precisión, rango de valores y frecuencia?
¿Los requisitos de Información indican los datos de salida del sistema incluyendo su destino, precisión, rango de valores, frecuencia y formato?
¿Los requisitos de Información indican los formatos de los reportes que generara el sistema?
¿Se identifican los requisitos de Información dentro de los requisitos funcionales?
¿Se identifican los requisitos de Información en requisitos separados de los Funcionales?

¿Los requisitos de Información tienen trazabilidad con requisitos Funcionales, Objetivos y Restricciones?
Conciso
¿Los requisitos son fáciles de leer y entender?
¿Los requisitos presentan una descripción clara dentro del contexto del problema?
¿La redacción de los requisitos es simple y clara para las personas que vayan a consultarlos?
¿Los requisitos están redactados en un lenguaje comprensible para el lector y no en un lenguaje de tipo técnico y especializado?
Ambigüedad
¿Los requisitos están sujetos a una única interpretación?
¿El lenguaje usado en la definición de los requisitos evita confusiones en el lector?
¿Existe un glosario de términos que permita definir los conceptos que se manejan en los requisitos y que habitualmente poseen más de una interpretación?
¿Los requisitos expresan hechos objetivos y no opiniones subjetivas?
Completo
¿Los requisitos proporcionan la información suficiente para su comprensión y No requieren ampliar detalles en su redacción?
¿Los requisitos están orientados a suplir las necesidades de los clientes y/o usuarios?
¿Los requisitos transmiten suficiente conocimiento para empezar o continuar con las actividades de diseño?
¿Se ha especificado el comportamiento del sistema para todos los tipos de errores más frecuentes y conocidos? (Ejemplo: años bisiestos, fechas invalidas...)
¿Los requisitos son suficientes? (i.e., podrían ser enviados a una empresa de desarrollo y tienen una probabilidad razonable de producir el producto que se quiere obtener)
¿Los requisitos contienen en sí mismos toda la información necesaria, y no remiten a otras fuentes externas que los expliquen con más detalle?
Consistente
¿Dentro del documento se omiten requisitos que puedan contradecir o estar en conflicto con cualquier otro requisito?
¿Existe una unión y/o relación adecuada de todos los requisitos para formar un todo (Sistema)?
¿Se omiten restricciones o descripciones contradictorias de una misma función del sistema?
Redundante
¿Se omiten requisitos que se encuentran expresando lo mismo?
¿Se omiten requisitos duplicados?
¿Dentro de los requisitos se omite la repetición innecesaria de una palabra, expresión o idea?
Preciso

¿Los requisitos hacen uso de valores numéricos para precisar las características del sistema?
¿En los requisitos se evita presentar detalles de diseño e implementación como datos técnicos, sentencias SQL, vistas, nombre de componentes, campos de bases de datos, objetos de software, entre otros?
¿En los requisitos han sido identificados los rangos esperados para todos los valores de entrada?
¿En los requisitos han sido incluidas las respuestas a todos valores de entrada tanto válida como inválida?
Modificable
¿Los requisitos presentan una estructura modificable que permite realizar cambios fáciles de introducir, manteniendo la estructura inicial del conjunto y continua siendo conciso, No ambiguo, completo y consistente?
Clasificación
¿Se identifica la importancia de cada uno de los requisitos para establecer prioridades a la hora de abordar el desarrollo?
¿La omisión de un requisito provoca una deficiencia en el sistema a construir?
Trazable
¿Los requisitos pueden ser referenciados o asociados fácilmente a los componentes de diseño y/o de implementación?
¿Todas las necesidades explícitas en el documento de requisitos están contenidas o asociadas en los casos de uso de negocio?
¿Se tiene especificada la matriz de Requisitos vs. Necesidades?
¿Se tiene especificada la matriz de Requisitos vs. Reglas de negocio?
Verificable
¿Los requisitos están expresados de manera que puedan comprobarse en un tiempo y a un costo razonable?
¿Los requisitos son un objetivo realista, posible de ser alcanzados con el dinero, el tiempo y los recursos disponibles?
Usabilidad
¿Dentro de los requisitos se indica que el sistema podrá ser usado por personas que pueden o no, tener habilidades en el trabajo con la computadora, debido a que se identifica que estará estructurado de forma sencilla?
¿La presentación y apariencia del sistema está especificada en los requisitos? (Combinaciones de colores, Tipo y tamaño de letra, estándares y Criterios generales de diseño)
¿Los requisitos indica que el sistema debe ser fácil de usar?
¿Los requisitos indican el idioma de la interfaz y de los mensajes del sistema?
Seguridad
¿Los requisitos reflejan patrones de seguridad teniendo en cuenta la alta sensibilidad de la información que se maneja?
¿El nivel de seguridad está especificado en los requisitos?

¿Se cuenta con un control de autenticación de acceso al sistema para que solo permita ingresar al personal autorizado?
¿Se cuenta con permisos de acceso a módulos, funciones, proceso y/o informes según perfiles de usuario?
¿Los requisitos contemplan las políticas, normas y estándares de la compañía, en cuanto a la seguridad requerida por el sistema?
¿Los requisitos indican el manejo de almacenamiento cifrado para información de acceso al sistema?
Eficiencia
¿El tiempo de respuesta o latencia esperado para la ejecución de los procesos generales están especificados en los requisitos?
¿El volumen de datos está especificado en los requisitos?
¿Está especificado en los requisitos el tiempo de procesamiento, transferencia de datos y el rendimiento del sistema?
¿Dentro de los requisitos se indica que el sistema debe ser rápido a la hora de procesar la información y dar respuesta a las peticiones de los usuarios?
¿En los requisitos se indica la planeación de secuencias de procesos del sistema (procesos ETL) para mantener la calidad del servicio en términos de tiempos de respuesta?
¿En los requisitos se indica la tasa esperada de velocidad de respuesta cuando el sistema sea utilizado por múltiples usuarios simultáneamente?
Disponibilidad
¿El tiempo que el sistema debe estar en estado disponible para proporcionar el servicio a los usuarios, está especificado en los requisitos?
El tiempo aceptable (máximo) del sistema fuera de línea está especificado en los requisitos?
¿Los requisitos indican planes de recuperación de caídas del sistema, como la utilización de sistemas de respaldo?
Fiabilidad
¿Se identifican en los requisitos atributos relacionados con la capacidad del sistema para mantener su nivel de prestación de servicio bajo condiciones establecidas durante un período?
¿La capacidad del sistema para tolerar errores está especificada en los requisitos?
¿La capacidad del sistema para recuperarse de los errores está especificada en los requisitos?
¿La capacidad del sistema para tolerar sobrecargas en el volumen de información, de usuarios o de procesos está especificada en los requisitos?
Mantenibilidad
¿Se identifican en los requisitos atributos relacionados con la facilidad de extender, modificar o corregir errores en el sistema?

¿El mantenimiento del sistema está especificado, incluyendo la capacidad de responder a los cambios en el entorno operativo, las interfaces con otros programas, precisión, rendimiento y capacidades adicionales de predecir?
¿Los requisitos indican la capacidad para realizar revisiones y cambios sobre una funcionalidad del sistema, de manera que no represente una exagerada inversión en recursos, el desarrollo de un cambio?
Portabilidad
¿Se identifican en los requisitos atributos relacionados con la capacidad del sistema para ser transferido desde una plataforma tecnológica a otra?
¿Los requisitos indican bajo qué sistemas operativos debe funcionar el sistema?
¿Los requisitos indican que el sistema debe garantizar compatibilidad con navegadores de uso común?
¿Los requisitos indican que el sistema debe garantizar capacidad de operar en arquitectura hardware de 32 ó 64 bits?

Artefacto para la etapa de diseño con la lista de chequeo:

- Lista de Chequeo Requisitos General.xls (sirve para coger todos los requisitos y abordarlos en conjunto)
- Lista de Chequeo Requisitos x Requisitos.xls (consiste en verificar cada requisito existentes en el alcance, se debe verificar uno a uno todas las validaciones que tienen esta plantilla)

El analista de pruebas está en la libertad de seleccionar que tipo de plantilla va a utilizar, la escogencia de esta plantilla depende del tiempo estimado y la complejidad de la prueba, por ejemplo si estimo un buen tiempo y la complejidad del proyecto es alta se debe verificar los requisitos con el artefacto “Lista de Chequeo Requisitos x Requisitos.xls y si cuenta con tiempos cortos y la complejidad de la prueba es baja o media se debe utilizar el artefacto “Lista de Chequeo Requisitos General.”.

2.4. Ejecución

Esta fase consiste en ejecutar los casos de prueba diseñados o ítems establecidos en la lista de chequeo para un ciclo, contrastando el resultado esperado de los requisitos con la realidad de lo especificado.

Dentro de la fase de ejecución se presentan las siguientes actividades:

2.4.1. Ejecución – Ciclo 1²⁴

Esta actividad consiste en ejecutar los casos de prueba diseñados o ítems establecidos en la lista de chequeo para el ciclo 1, con el fin de comprobar que cada uno de los requisitos documentados, corresponden a las necesidades de los clientes y/o usuarios (validación) y comprobar que la especificación cumple con los criterios de calidad oportunos (verificación).

El resultado obtenido durante la ejecución del ciclo 1 debe ser registrado, al igual, que todas las evidencias de la ejecución.

- Los pasos para ejecutar esta actividad son los siguientes:
 4. Establecer los casos de prueba diseñados o ítems de las listas de chequeo para el ciclo 1, que van hacer ejecutados.
 5. Ejecutar los casos de prueba o ítems de las listas de chequeo.
 6. Registrar los resultados de cada caso de prueba o ítem de las listas de chequeo.
 7. Tomar evidencias de lo ejecutado.
- Artefactos de Entrada:
 - ✓ Plantilla de Diseño de Casos de Prueba.
 - ✓ Plantilla de Lista de Chequeo.
- Artefactos de Salida:
 - ✓ Plantilla de Diseño de Casos de Prueba con resultados.
 - ✓ Plantilla de Lista de Chequeo con resultados.
 - ✓ Archivo de Evidencias.
- Responsable:
 - ✓ Analista de Pruebas

¿Qué se debe Probar?

1. Requisitos Funcionales, da tal manera que estos describan la funcionalidad requerida de un sistema, lo que el sistema debe hacer, función que debe ejecutar y

²⁴ **Ciclo1:** Ejecución de los casos de prueba o listas de chequeo en la primera versión de los requisitos recibida.

que permite cumplir o están direccionados a cumplir una necesidad u objetivo de clientes y/o usuario.

Para que los requisitos funcionales tengan un alto nivel de calidad deben cumplir cada una de las siguientes características:

- **Conciso:** Un requisito es conciso si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarlo en un futuro.
- **No ambiguo:** Se considera que un requisito es ambiguo cuando puede ser interpretado de formas diversas por diferentes personas. Así, el estándar IEEE 830 establece que un requisito es No-ambiguo “si, y sólo si, puede estar sujeto a una única interpretación.”

El lenguaje usado en la definición del requisito no debe causar confusiones al lector.

- **Completo:** Un requisito está completo si no necesita ampliar detalles en su redacción, es decir, si se proporciona la información suficiente para su comprensión.
- **Verificable:** Un requisito es verificable tienen un objetivo realista, posible de ser alcanzados con el dinero, el tiempo y los recursos disponibles.

Se considera que un requisito es verificable “si existe un proceso acotado (en plazo y presupuesto) que permita determinar que el sistema construido satisface lo descrito en el propio requisito”.

- **Consistente:** Un requisito es consistente si no es contradictorio con otro requisito (No existan contradicciones entre los requisitos).

Se considera que un requisito es consistente “si, y sólo si, no hay ningún otro requisito descrito dentro de un documento de especificación de requisitos que esté en conflicto con cualquier otro.”

- **No Redundante:** Un requisito es redundante si se encuentran requisitos expresando lo mismo.

Se considera que un requisito No es redundante “si, y sólo si, no hay ningún otro requisito que sea igual dentro de un documento de especificación de requisitos.”

- **Preciso:** Un requisito es preciso si hace uso de valores numéricos para precisar las características del sistema.

La precisión es aplicable, ante todo, a los requisitos no funcionales. Por ejemplo, no es útil decir “El tiempo de respuesta será más bien rápido”, sino “El tiempo de respuesta será menor que dos segundos”.

- **Clasificación:** No todos los requisitos son igual de importantes. Los requisitos pueden clasificarse por diversos criterios:
 - **Importancia:** Pueden ser esenciales, condicionales u opcionales. La clasificación de los requisitos por orden de importancia es una excelente práctica que permite establecer prioridades a la hora de abordar el desarrollo.

Esta clasificación por el atributo “importancia” es necesaria desde un punto de vista práctico, dado que la construcción de sistemas se hace con un presupuesto limitado y en un plazo limitado.
 - **Estabilidad:** Cambios que pueden afectar al requisito. En el entorno de gestión, las necesidades del mercado imponen cambios continuos sobre los sistemas software, que se traducen, a su vez, en cambios en los requisitos.
 - **Necesario:** Un requisito es necesario si su omisión provoca una deficiencia en el sistema a construir, y además su capacidad, características físicas o factor de calidad no pueden ser reemplazados por otras capacidades del producto o del proceso.
- **Modificable:** Se considera que un requisito es modificable “si su estructura es tal que permite realizar cambios sobre el requisito, manteniendo la estructura inicial del conjunto y continua siendo conciso, No ambiguo, completo y consistente. Los cambios son fáciles de introducir.
- **Trazable:** Se considera que un requisito es trazable, si está claro el origen de cada requisito (quién o qué lo pide). Además, que todas las necesidades explícitas en el documento de requisitos estén contenidas en los casos de uso de negocio; es decir, que los requisitos se vean reflejados en al menos un caso de uso y que cada caso de uso corresponda a un requisito

2. Los requisitos No Funcionales tienen que ver con las calidades sistémicas que el sistema debe cumplir. Son las condiciones externas a la funcionalidad que aun que

no las afectan en su objetivo, pueden si afectar la percepción sobre la ejecución de la misma.

Para que los requisitos No funcionales tengan un alto nivel de calidad deben cumplir cada una de las siguientes condiciones:

- **Fiabilidad:** Conjunto de atributos relacionados con la capacidad del software de mantener su nivel de prestación bajo condiciones establecidas durante un período establecido (Madurez, Recuperabilidad y Tolerancia a fallos)
 - **Usabilidad:** Requisitos que determinan las características generales de la capa de presentación del sistema en cuanto a las características de diseño gráfico de la misma, además de las facilidades para que el uso del sistema por parte del usuario final.
 - **Eficiencia:** Conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesarios bajo condiciones establecidas (Comportamiento en el tiempo y Comportamiento de recursos)
 - **Mantenibilidad:** Requisitos relacionados con la capacidad para realizar revisiones y cambios sobre la funcionalidad del sistema de manera que no represente una exagerada inversión en recursos el desarrollo de un cambio.
 - **Portabilidad:** Conjunto de atributos relacionados con la capacidad de un sistema software para ser transferido desde una plataforma a otra (Capacidad de instalación, Capacidad de reemplazamiento, Adaptabilidad y Co-Existencia)
 - **Disponibilidad:** Conjunto de atributos relacionados con la capacidad del sistema para estar disponible para los usuarios, esto se refleja en el tiempo estimado, esperado y requerido por el usuario para que el sistema esté disponible.
3. Reglas de Negocio: Son todas aquellas restricciones, leyes, límites y cualquier otra condición que puede restringir, condicionar o afectar la operación de un requisito funcional.
4. Requisitos de Información: Son cada uno de los datos que serán almacenados, presentados, convertidos, eliminados por la ejecución de un requisito funcional.

2.4.2. Reportar Errores en Bugtracker

Esta actividad consiste en registrar en el bugtracker las incidencias encontradas durante la ejecución de los casos de prueba diseñados o ítems establecidos en la listas de chequeo.

Al reportar una incidencia debe quedar claro si es un Error, Pregunta, Sugerencia o Hallazgo y estar acompañado de una valorización de severidad y prioridad.

- Los pasos para ejecutar esta actividad son los siguientes:
 1. Registrar el fallo en el caso de prueba o No cumplimiento del ítem en la lista de chequeo.
 2. Registrar la incidencia en el bugtracker (Nro. Incidencia, Nombre Requisito, Tipo de Requisito, Tipo Incidencia, Severidad, Prioridad, Descripción y Estado Incidencia).
 3. Asociar en el caso de prueba fallido o en el ítem que no se cumplió de la lista de chequeo, el número de la incidencia generada en el bugtracker para hacer monitoreo y control.
- Artefactos de Entrada:
 - ✓ Plantilla de Diseño de Casos de Prueba.
 - ✓ Plantilla de Lista de Chequeo.
- Artefactos de Salida:
 - ✓ Plantilla de Diseño de Casos de Prueba con resultados.
 - ✓ Plantilla de Lista de Chequeo con resultados.
 - ✓ Archivo de Evidencias.
 - ✓ Plantilla de Bugtracker.
- Responsable:
 - ✓ Analista de Pruebas

2.4.3. Prueba de Confirmación y Regresión

Esta actividad consiste en asegurar (Validar y Verificar) que cada uno de los casos de prueba que fallaron o los ítems que los requisitos no cumplieron, hayan sido modificados o corregidos satisfactoriamente sin generar o introducir nuevas incidencias. La corrección de las incidencias generadas en el bugtracker (Errores, Preguntas, Sugerencias o Hallazgos) o incorporación de nuevos requisitos, no deben de afectar los requisitos que venían comportándose correctamente.

- Los pasos para ejecutar esta actividad son los siguientes:
 1. Identificar los casos de prueba diseñados o ítems de las listas de chequeo que van hacer ejecutados.
 2. Ejecutar los casos de prueba o ítems de las listas de chequeo.
 3. Registrar los resultados de cada caso de prueba o ítem de las listas de chequeo en un nuevo ciclo.
 4. Tomar evidencias de lo ejecutado.
 5. Cerrar o reabrir las incidencias generadas en el bugtracker.
- Artefactos de Entrada:
 - ✓ Plantilla de Diseño de Casos de Prueba.
 - ✓ Plantilla de Lista de Chequeo.
 - ✓ Plantilla de Bugtracker.
- Artefactos de Salida:
 - ✓ Plantilla de Diseño de Casos de Prueba con resultados.
 - ✓ Plantilla de Lista de Chequeo con resultados.
 - ✓ Archivo de Evidencias.
 - ✓ Plantilla de Bugtracker.
- Responsable:
 - ✓ Analista de Pruebas

2.5. Aceptación

Esta fase consiste en comprobar que los clientes y/o usuarios están de acuerdo con la especificación de los requisitos y así poder dar por finalizado el proceso de las pruebas por medio de una carta de aceptación.

Dentro de la fase de ejecución se presentan las siguientes actividades:

2.5.1. Aprobación de los Requisitos

Esta actividad consiste en que los clientes y/o usuarios acepten y den por buenas las mejoras y correcciones realizadas a los requisitos, debido a que estos actores son quienes en realidad conocen lo que se requiere para suplir las necesidades.

- Los pasos para ejecutar esta actividad son los siguientes:
 1. Solicitar la Aprobación de los requisitos.
 2. Enviar la respectiva documentación a los actores que deben aprobar los requisitos.
 3. Recibir por medio de algún medio la aprobación de los requisitos.
- Artefactos de Entrada:
 - ✓ Documento final de requisitos.
- Artefactos de Salida:
 - ✓ Documento final de requisitos.
 - ✓ Aprobación de los requisitos.
- Responsables:
 - ✓ Analista de Pruebas.
 - ✓ Clientes y/o usuarios.

2.5.2. Análisis de Rechazo

Cuando exista un rechazo por parte los clientes y/o usuarios a las mejoras y correcciones realizadas a los requisitos, es en esta actividad donde se determina si la

causal de la no aprobación corresponde a un defecto no detectado en la fase de ejecución o un cambio en el detalle del requisito por desconocimiento del negocio.

- Los pasos para ejecutar esta actividad son los siguientes:
 1. Recibir la no aprobación de los requisitos.
 2. Identificar el o los requisitos que ocasionaron el rechazo.
 3. Determinar la causal de rechazo.
 4. Solicitar corrección o reportar en bugtracker.
- Artefactos de Entrada:
 - ✓ Rechazo de los requisitos.
 - ✓ Documento de requisitos.
- Artefactos de Salida:
 - ✓ Plantilla de Carta de aceptación de los requisitos.
 - ✓ Documento corregido de requisitos.
 - ✓ Plantilla de Bugtracker.
- Responsable:
 - ✓ Analista de Pruebas.

2.5.3. Carta de Aceptación.

En esta actividad se hace constar por escrito que los requisitos cumplen con todas las características y atributos de calidad para proceder a la siguiente fase del ciclo de desarrollo de Software.

- Los pasos para ejecutar esta actividad son los siguientes:
 5. Elaborar carta de aceptación.
 6. Enviar al equipo de trabajo la carta de aceptación.
 7. Dar por terminada la prueba de requisitos.
- Artefactos de Entrada:

- ✓ Aprobación de los requisitos.
- Artefactos de Salida:
 - ✓ Plantilla de Carta de aceptación de los requisitos.
 - ✓ Documento final de requisitos.
- Responsable:
 - ✓ Analista de Pruebas.

CONCLUSIONES

Con este trabajo se concluye que las pruebas de requisitos hace parte del proceso de desarrollo de software, es un puente muy importante para las otras etapas ya que da el aseguramiento de la calidad en la primera etapa, como también es importante el diseño, la implementación, la validación y el mantenimiento del sistema dentro de la ingeniería de software.

Esto significa que las pruebas de requisitos garantizan el desarrollo de un producto final. El mundo de las pruebas sabe y tiene claro que uno de los mayores problemas para un ingeniero de sistemas es la elicitación y especificación de los requisitos, ya que por un lado busca resolver las diferencias de la comunicación (ambigüedad), culturales y técnicas que enfrentan los usuarios y desarrolladores con un limitante del tiempo en búsqueda de obtener las necesidades y restricciones del sistema; y por otro lado la búsqueda de técnicas, métodos, y herramientas que permitan obtener una especificación de los requisitos entendible tanto para los ingenieros que están involucrados en el diseño como también para los clientes que son los que realmente saben las necesidades de sistema.

Por las razones anteriormente se creó un proceso de calidad las pruebas dentro de la ingeniería de software, para aplicar a los requisitos y con eso disminuimos costos para todo el proyecto, errores de estructura y diseño en los requisitos, errores de diseño en los casos de uso, y lo más importante se eliminan posibles errores que se pueden encontrar en otra etapa, donde el costo de la solución aumentaría y por ultimo aumenta la probabilidad que se terminen los proyecto con éxito y con un aseguramiento de la calidad. En Colombia son pocas las empresas que tienen este proceso implementado en sus procesos de calidad ya que no le ven la importancia a la calidad. Este proceso de calidad tiene una gran importación para cualquier empresas y se tiene que demostrar, esta problemática se concluye que mucha parte se debe a lo cultural de Colombia, apenas en estos últimos años en Colombia ya hay varias empresas que han implementado dentro de sus procesos , este proceso de pruebas y se han dado cuenta con hechos e indicadores que gracias a las pruebas hoy en día tienen procesos con calidad,

han eliminado los reprocesos y han bajado los costos en sus proyectos y tienen unas empresas más competitivas en el mercado.

Es necesario que todas las empresas mejoren sus procesos de calidad, si desean ser competitivas en el mercado con sistemas óptimos y de buena calidad, las empresas del exterior exigen altos estándares de calidad que la mayoría de las empresas en Colombia no pueden satisfacer en la actualidad. Para acabar esta problemática las empresas desarrolladoras de software deben ser las primeras en vender sus aplicativos con certificaciones de calidad por empresas que prestan el servicio de software, así estamos logrando que nuestras empresas compren, alquilen, y vendan aplicativos con altos estándares de calidad.

Se creó de un proceso de pruebas de requisitos en proyectos de software ágil, donde los se demuestran tiempos cortos y se verifican y se validan los requisitos de software, este proceso está compuesto por las siguientes etapas dentro de las pruebas que son: Planeación, Diseño donde se tienen dos propuestas que son la creación de casos de prueba y una lista de chequeo para tiempos más cortos, luego ejecución y por último se le entrega al cliente una carta de aceptación.

Se llega a la conclusión que es difícil proponer una técnica, método, metodología o proceso que cubra y asegure la calidad a los sistemas, esto se debe a que cada sistema es diferente, sin importar que estén desarrollados bajo el mismo ambiente, además de las diferentes características de los usuarios y clientes, hay clientes que quieren que estos procesos sean parametrizables es decir que se ajusten como ellos lo desean. Por lo que nos pone a pensar que este proceso se debe seguir al pie de la letra y se debe procurar no modificarlo, se tiene que tener en cuenta que para lograr una buena calidad en nuestros sistemas todas las etapas dentro de un ciclo de vida de desarrollo de software tienen que aplicárseles pruebas de software, desde la etapa temprana que es la especificación de los requisitos como la etapa final que son las pruebas funcionales o pruebas de aceptación con todo este seguimiento a lo largo del desarrollo del sistema estamos garantizando un buen sistema con los más altos estándares de calidad, esto es un trabajo de todos los ingenieros de sistemas que conforman el equipo de trabajo para un proyecto, si todos somos organizados y aseguramos que en cada etapa del ciclo de vida del software hay pruebas, estamos construyendo un aplicativo con cero errores lo que significa y asegura la calidad del sistema.

Un buen proceso de pruebas en requisitos nos facilitara en un futuro del desarrollo del software el cumplimiento de las expectativas o necesidades del cliente, esto evitará sobrecostos de implementación, generara en menor proporción reprocesos y aportará a la calidad del aplicativo desarrollado.

La detección de un error en etapas tempranas del proceso de software es menos costosa, por lo que las pruebas de requisitos toman mayor importancia, debido a que se efectúan o se aplican en una de las primeras etapas del ciclo de desarrollo del software y así en etapas posteriores se previene la generación de errores por malas especificaciones de los requisitos.

GRUPO DE TRABAJO

Estudiantes:

Nombre: Juan Camilo Hernández González

E-mail: jherna15@eafit.edu.co

Código: 200210173010

Celular: (300) 3671850

Nombre: John Felipe Morales Isaza

E-mail: jmoral13@eafit.edu.co

Código: 200310052010

Celular: (300) 3055672

Asesora:

Nombre: NATALIA ESCOBAR BLANCO

E-mail: nescoba6@gmail.com

Teléfono: 4129182

Celular: (300) 6544889

BIBLIOGRAFÍA

- Universidad EAFIT, Centro Cultural Luís Echavarría Villegas. GUÍA PARA LA PRESENTACIÓN DE PROYECTOS Y TESIS DE GRADO, Medellín, 2007.
- Better Software, To Track or Not to Track, Lisa Crispin.
Internet: <http://lisacrispin.com/downloads/TrackOrNot.pdf>. Junio de 2007.
- El modelo del negocio como base del modelo de Requisitos1
Internet:<http://docencia.udea.edu.co/ingenieria/ArquitecturaSoftware/documentos/Del%20Modelo%20Del%20Negocio%20Al%20Modelo%20De%20Requisitos.pdf>
- Ingeniería de software pruebas.
Internet: <http://www.ingenierosoftware.com/pruebas/gestionerrores.php>
- InformIT, Testers: The Hidden Resource, Lisa Crispin y Janet Gregory.
Internet: <http://www.informit.com/articles/article.aspx?p=1322400> , Febrero de 2009.
- Better Software, The Tester Who Came In from the Cold, Lisa Crispin.
Internet: <http://lisacrispin.com/downloads/CrispinFINALTester.pdf>. Octubre de 2008.
- Durán Toro, Amador. Un Entorno Metodológico de Ingeniería de requisitos.
Internet:<http://fondosdigitales.us.es/tesis/tesis/30/un-entorno-metodologico-de-ingenieria-de-requisitos-para-sistemas-de-informacion/>
- Ingeniería De Requisitos Ingeniería De Software.
<http://www.monografias.com/trabajos6/resof/resof.shtml>
- Curso de Ingeniería de Requisitos, dictado en Febrero 5 – Marzo 19, por el Ingeniero Faber López.

- Requerimientos vs. Requisitos
<http://asprotech.blogspot.com/2011/03/requerimientos-vs-requisitos.html>
- Monzón Antonio. Calidad de la Especificación.
Documento: Calidad_de_la_Especificacion.pdf