



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

ESTUDIO DE FACTIBILIDAD DE UN CENTRO DE PRUEBAS DE SOFTWARE EN GUATEMALA

Carlos Alejandro Pérez Ventura

Asesorado por el Ing. Edgar Estuardo Santos Sutuj

Guatemala, octubre de 2008

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**ESTUDIO DE FACTIBILIDAD DE UN CENTRO DE
PRUEBAS DE SOFTWARE EN GUATEMALA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR:

CARLOS ALEJANDRO PÉREZ VENTURA
ASESORADO POR EL ING. EDGAR ESTUARDO SANTOS SUTUJ

AL CONFERÍRSELE EL TÍTULO DE
INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, OCTUBRE DE 2008

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Inga. Glenda Patricia García Soria
VOCAL II	Inga. Alba Maritza Guerrero Spínola de López
VOCAL III	Ing. Miguel Ángel Dávila Calderón
VOCAL IV	Br. Milton De León Bran
VOCAL V	Br. Isaac Sultán Mejía
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Fredy Javier Gramajo López
EXAMINADOR	Inga. Virginia Victoria Tala Ayerdi
EXAMINADOR	Ing. César Augusto Fernández Cáceres
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas

HONORABLE TRIBUNAL EXAMINADOR

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

ESTUDIO DE FACTIBILIDAD DE UN CENTRO DE PRUEBAS DE SOFTWARE EN GUATEMALA,

tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, en julio de 2007.

f. 

Carlos Alejandro Pérez Ventura

Guatemala, 12 de Septiembre del 2008

Ing. Carlos Azurdia
Coordinador de Tesis
Facultad de Ingeniería
Escuela de Ciencias y Sistemas

El motivo de la presente es para informarle que he revisado el trabajo de graduación del alumno **Carlos Alejandro Pérez Ventura**, titulado "Estudio de factibilidad de un Centro de Pruebas de Software en Guatemala", a mi parecer cumple con los requisitos planteados como trabajo de tesis.

Atentamente,

A handwritten signature in black ink, consisting of several loops and a long horizontal stroke, positioned above the printed name.

Ing. Edgar Santos
Colegiado-6266
Revisor de Trabajo de Graduación



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 22 de Septiembre de 2008

Ingeniero
Marlon Antonio Pérez Turk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **CARLOS ALEJANDRO PEREZ VENTURA** titulado: **"ESTUDIO DE FACTIBILIDAD DE UN CENTRO DE PRUEBAS DE SOFTWARE EN GUATEMALA"**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme.

Atentamente,


Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación



UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
TEL: 24767644

E
S
C
U
E
L
A

D
E

C
I
E
N
C
I
A
S

S
I
S
T
E
M
A
S

El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, de trabajo de graduación titulado "ESTUDIO DE FACTIBILIDAD DE UN CENTRO DE PRUEBAS DE SOFTWARE EN GUATEMALA" presentado por el estudiante CARLOS ALEJANDRO PEREZ VENTURA, aprueba el presente trabajo y solicita la autorización del mismo.

"ID Y ENSEÑAD A TODOS"


Ing. Marlon Antonio Pérez Turk



Director, Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 21 de octubre 2008



El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **ESTUDIO DE FACTIBILIDAD DE UN CENTRO DE PRUEBAS DE SOFTWARE EN GUATEMALA**, presentado por el estudiante universitario **Carlos Alejandro Pérez Ventura**, procede a la autorización para la impresión del mismo.

IMPRIMASE.

Ing. Murphy Olimpo Paiz Recinos
DECANO



Guatemala, Octubre de 2008

/gdech

AGRADECIMIENTOS A:

- DIOS** Por darme la vida y la sabiduría para tomar las decisiones correctas en la vida
- LA VIRGEN MARÍA** Por su protección y guía en las pruebas más difíciles que se han presentado
- MIS PADRES** Por el esfuerzo realizado para tener la oportunidad de estudiar y así culminar mis estudios
- MI FAMILIA** A mis hermanos, por el apoyo brindado. A mis primos, por el ejemplo a seguir. A mis tíos y tías, por las palabras de aliento durante toda la carrera. A Sasha, por la compañía y ayuda brindada.
- MIS AMIGOS** A los viejos amigos del colegio que siempre me han apoyado, y con quienes compartimos los mejores momentos de la Universidad
- MIS CATEDRÁTICOS** Por compartir el conocimiento sin egoísmo. Al Ing. Edgar Santos, por los consejos y ayuda al desarrollo de este trabajo

DEDICATORIA A:

- DIOS** Porque todo lo que hago es en nombre de Dios. Esta meta alcanzada la pongo en manos de Dios, para que me permita alcanzar las siguientes
- MIS PADRES** Estoy orgulloso que Alejandro Pérez y Rosa Amalia Ventura hayan sido elegidos para ser las personas que guíen mi vida
- MIS HERMANITOS** Claudia María, Javier Eduardo, Ángel Miguel, Andrés Emilio y Darío Andrés, para que tengan un ejemplo más de superación y lucha por alcanzar las metas propuestas

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	V
GLOSARIO	VII
RESUMEN	XIII
OBJETIVOS	XV
INTRODUCCIÓN.....	XVII
1. NATURALEZA DEL PROYECTO	1
1.1 Caso de estudio	1
1.1.1 Descripción general del Centro de ensayos de Software.....	1
1.1.2 Servicios y productos	2
1.2 <i>Benchmarking</i>	5
1.3 Descripción del Centro de Pruebas (Guatemala).....	10
1.4 Productos y servicios	14
1.4.1 Pruebas funcionales.....	15
1.4.2 Pruebas no funcionales.....	16
1.4.3 Pruebas estructurales	19
1.5 Herramientas de apoyo.....	21
1.5.1 Herramientas de soporte para la administración de las pruebas y el test.....	22
1.5.2 Herramientas de soporte para las pruebas estáticas.....	25

1.5.3	Herramientas de soporte para la especificación de las pruebas.	27
1.5.4	Herramientas de soporte para la ejecución de las pruebas y el log.....	29
1.5.5	Herramientas de soporte para desempeño y monitoreo	33
1.6	Nivel de pruebas en Guatemala	35
1.6.1	Generales	36
1.6.2	De los tipos de pruebas	41
1.6.3	De las herramientas.....	47
2.	MERCADO.....	55
2.1	Estudio de mercado.....	55
2.1.1	Tamaño del mercado.....	55
2.1.2	Consumo aparente	56
2.1.3	Demanda potencial histórica.....	57
2.1.4	Demanda insatisfecha	57
2.1.5	Objetivos del estudio de mercado.....	57
2.1.6	Resultados obtenidos	58
2.1.7	Conclusiones del estudio realizado	70
2.2	Promoción del producto o servicio.....	71
2.3	Fijación y políticas de precio.....	71
2.4	Riesgos y oportunidades del mercado.....	72
3.	PRODUCCIÓN.....	73
3.1	Descripción del proceso productivo o prestación del servicio.....	73
3.1.1	Las pruebas como proceso	74
3.1.2	Proceso paralelo e integrado con ciclo de vida.....	77
3.2	Características tecnológicas	78
3.3	Equipo e instalaciones.....	81

3.4	Recursos adicionales	82
3.5	Procedimientos de mejora continua	83
3.6	Impacto social	83
4.	ORGANIZACIÓN	85
4.1	Modelo de trabajo	85
4.1.1	Modelo vertical	85
4.1.2	Modelo horizontal	86
4.1.3	Modelo de servicio	86
4.2	Estructura organizacional.....	87
4.3	Objetivos y funciones del puesto	88
4.4	Captación y desarrollo del personal.....	89
4.5	Administración de sueldos y salarios	89
5.	FINANZAS	93
5.1	Flujo de efectivo	93
5.2	Sistema de financiamiento	96
	CONCLUSIONES.....	97
	RECOMENDACIONES	101
	REFERENCIAS.....	105
	BIBLIOGRAFÍA.....	107
	APÉNDICES.....	109
	Encuesta I.....	109
	Encuesta II	113

Resultados tabulados	120
Gastos estimados	121

ÍNDICE DE ILUSTRACIONES

FIGURAS

1. Modelo comercial	9
2. Rol del Centro de Pruebas de Software	11
3. Objetivo de las pruebas de software	37
4. Porcentaje dedicado a pruebas de software	38
5. Participación en las pruebas de software	40
6. Pruebas funcionales	42
7. Pruebas a las características del software	43
8. Pruebas a la estructura del software	44
9. Pruebas de cambios al software.....	45
10. Administración de las pruebas de software	46
11. Herramientas para la administración de pruebas de software.....	48
12. Herramientas para pruebas estáticas.....	49
13.Herramientas para la especificación de pruebas.....	51
14. Herramientas para la ejecución de pruebas	52
15. Herramientas para monitoreo	53
16. Cantidad de clientes	58
17. Número de empleados	59
18. ¿Contrata servicios de control de calidad y pruebas de software?.....	60
19. Número de programadores.....	61
20. Personas por proyecto	62
21. Desarrollo de software a la medida	63
22. Nivel de implementación	64

23. Interés de ensayos de plataforma	66
24. Interés de servicios de pruebas	67
25. Control de calidad actual.....	69
26. Diagrama de flujo de las pruebas como proceso	74
27. Diagrama de flujo del proceso paralelo	77
28. Modelo vertical	85
29. Modelo horizontal	86
30. Modelo de servicio	87
31. Estructura Organizacional	88

GLOSARIO

ANS	Acuerdos de nivel de servicio. Es una herramienta que ayuda a los proveedores y clientes internos a llegar a un consenso en términos del <i>nivel de servicios</i> requeridos para sustentar las necesidades del negocio
Artefacto	Objeto de RUP
Aseguramiento de Calidad	Construcción de la calidad en el producto. <i>Quality Assurance</i> , ISO 8402, 1994) Todas las actividades planificadas y sistemáticas necesarias para aportar la confianza suficiente en que un producto o servicio cumplirá con unos requisitos dados de calidad. <i>(Those planned and systematic actions necessary to provide sufficient confidence that a product or service will satisfy given requirements for quality.)</i>
Bug	Es un mal funcionamiento de un elemento de software. Que un programa haga cosas no queridas, o que no haga las cosas que debería. Un error de software, es el resultado de un fallo de programación inducido durante el proceso de creación de programas de ordenador o computadora (software).

Calidad	(<i>Quality</i> , ISO 8402, 1994) Conjunto de propiedades y de características de un producto o servicio, que le confieren su aptitud para satisfacer unas necesidades explícitas e implícitas.
CASE	Computer-Aided Software Engineering es el uso de herramientas de software para la asistencia en el desarrollo y mantenimiento del software.
Caso de prueba	Conjunto de entradas, precondiciones para la ejecución y salidas esperadas desarrolladas con el objetivo de testear algo concreto del software (ejecutar un camino del programa en particular, verificar la conformidad de un requisito concreto, detectar tipos de errores específicos)
Control de calidad	Verificar la calidad del producto.
Defecto	La no ejecución de exigencias de uso intencionadas.
Demanda	Cantidad y calidad de bienes o servicios que los consumidores están dispuestos a comprar a un precio y unas condiciones dadas en un momento determinado.

Drivers	<i>(bottom-up)</i> : hace énfasis en la programación y pruebas tempranas, que pueden comenzar tan pronto se ha especificado el primer módulo. Este enfoque tiene el riesgo de programar cosas sin saber como se van a conectar al resto del sistema, y esta conexión puede no ser tan fácil como se creyó al comienzo. La reutilización de código es uno de los mayores beneficios del enfoque <i>bottom-up</i> .
Efectividad	Proceso cumpla bien su cometido.
Eficiencia	Cumpla su cometido optimizando los recursos utilizados.
Error	Una acción humana que puede producir resultados incorrectos.
Fallo	Una desviación del funcionamiento esperado.
Fiabilidad	Grado en que el sistema responde bajo las condiciones definidas durante un intervalo de tiempo dado. Se divide en las subcaracterísticas madurez, tolerancia a fallos, capacidad de recuperación.
Funcionalidad	Grado en que las necesidades asumidas o descritas se satisfacen. Se divide en las subcaracterísticas idoneidad, precisión, interoperabilidad, seguridad.

Gerencia de proyecto	Es el ente responsable de supervisar los horarios, los presupuestos y los recursos totales del proyecto.
Inconformismo	La no ejecución de exigencias especificadas.
ISTQB	International Software Testing Quality Board.
Mantenimiento	Esfuerzo requerido para implementar cambios. Se divide en las subcaracterísticas capacidad de ser analizado, cambiabilidad, estabilidad, facilidad de prueba.
Middleware	Es un software de conectividad que permite ofrecer un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas.
Modelo de crecimiento de fiabilidad del software	Gráficas que reflejan el patrón de defectos de un aplicativo de software basándose en datos obtenidos durante las fases formales de pruebas.
Oferta	Cantidad de bienes o servicios que los productores están dispuestos a ofrecer a un precio dado en un momento determinado
Plaza	Son los puntos de venta, canales de distribución y tiempos de entrega.

Plaza	El lugar en donde está disponible tu producto. Los profesionales de la mercadotecnia deben de saber que el método de distribución, igual que el precio, ha de ser compatible con la imagen de la marca. La distribución designa la forma en que el producto se pone a disposición del consumidor: dónde se distribuye, cómo se compra y cómo se vende.
Precio	Es lo que pagas para cualquier producto o servicio que consumes. Éste influye profundamente en las percepciones de la marca por parte del consumidor. Indica qué y cuánto el cliente paga por un producto. Las compañías aplican varias estrategias al fijarlo. Algunos productos compiten con el precio.
Producto	Es todo aquello que es susceptible de ser comprado, cambiado, traspasado, etc. Incluye la forma en que se diseña, se clasifica, se posiciona, se empaca y se reconoce a través de una marca.
Promoción	Son las actividades encaminadas a cambiar el comportamiento del consumidor, dándole algo a por ese cambio de comportamiento. La comunicación indica todas las comunicaciones relacionadas con el <i>marketing</i> que se dan entre vendedor y comprador. Las herramientas de este elemento son las de la mezcla promociona.

Publicidad	Crear un impacto directo sobre el cliente para que se compre el producto
Software	Se denomina software, programática, equipamiento lógico o soporte lógico a todos los componentes intangibles de un ordenador o computadora, es decir, al conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema.
Stubs	(<i>top-down</i>): enfatiza la planificación y conocimiento completo del sistema. Se entiende que la codificación no puede comenzar hasta que no se haya alcanzado un nivel de detalle suficiente, al menos en alguna parte del sistema. Esto retrasa las pruebas de las unidades funcionales del sistema hasta que gran parte del diseño se ha completado.
Test exitoso	Un <i>test</i> que descubre defectos
Testear software	Examinar un artefacto de software con la intención de encontrar defectos (tal que tus clientes no lo hagan).
Testware	Software que sirve para realizar pruebas al software. Cualquier resultado de los procesos de prueba (casos de <i>test</i> , planes, <i>scripts</i> , <i>stubs</i> , <i>drivers</i> , etc.).

RESUMEN

¿Por qué hacemos pruebas? Las dos razones principales son: para hacer un juicio acerca de la calidad y aceptabilidad, y para descubrir problemas. Probamos porque sabemos que errar es de humanos, esto es especialmente cierto en el ambiente del software y los sistemas controlados por software.

Objetivos de las pruebas

- Encontrar defectos
- Mejorar la calidad de los productos y aplicaciones
- Reducir los esfuerzos en mantenimiento
- Mejorar la satisfacción de los usuarios y clientes

Clasificación de las pruebas

Pruebas unitarias

Se realizan sobre componentes simples de forma aislada. Normalmente son realizadas por los propios programadores y no de forma sistemática. Normalmente se realizan pruebas de caja blanca debido a la falta de especificaciones claras a ese nivel.

Pruebas de integración

Son pruebas de acoplamiento de componentes unitarios. Se enfoca especialmente en la integración y en las interfaces. Se realizan estrategias *top-down* o *bottom-up*. Se necesitan drivers (*bottom-up*) y stubs (*top-down*).

Normalmente estas pruebas las realizan los programadores aunque a veces se tienen grupos especializados de integración, todo dependiendo del desarrollo del proyecto.

Pruebas funcionales

Habitualmente se realizan pruebas de caja negra y suelen complementarse con pruebas de caja blanca. Es recomendable que estas pruebas las realicen equipos independientes. Son más efectivas si van paralelas al desarrollo debido a la mejora de la calidad en los requisitos.

Pruebas no funcionales

Son pruebas muy dependientes del sistema o aplicación y los requisitos no funcionales particulares. Un son las pruebas de carga en donde se validan los requisitos de prestaciones del sistema o aplicación.

Pruebas de aceptación

Son realizadas por el usuario. Pueden llegar a ser tan formales como las del sistema, una parte de estas o las que el usuario considere.

Pruebas de regresión

Estas pruebas son cualquiera de las anteriores, únicamente que son aplicadas a las partes que no son afectadas por los cambios realizados.

Pruebas de campo

Estas son las pruebas básicas que se realizan al sistema una vez que está en producción.

OBJETIVOS

- **General:**

Evaluar la factibilidad de creación de un centro que permita a los desarrolladores de software tanto individuales como colectivos realizar pruebas de varios tipos, a fin de certificar la calidad del mismo.

- **Específicos:**

1. Identificar los diferentes tipos de pruebas al software que ayudan al aseguramiento de la calidad.
2. Evaluación de la aplicación de estos tipos de pruebas en el software que se produce y utiliza en Guatemala.
3. Servir de base para ser comparado con otros países que ya poseen o desean crear un centro de pruebas.
4. Demostrar la factibilidad de abrir un centro de pruebas para software en el país.

INTRODUCCIÓN

En el presente trabajo de investigación se ha realizado un estudio de factibilidad para instalar un Centro de Pruebas de Software en Guatemala. Este es un estudio preliminar, ya que en Guatemala existen entidades interesadas en crear este tipo de centros en los próximos años.

En la primera parte del trabajo, se describe al Centro de Pruebas, se inicia con un caso de estudio acerca del Centro de Ensayos de Software ubicado en Uruguay, el cual es básicamente el mismo tipo de centro que se busca instalar en el país. Luego del caso de estudio se define al Centro de Pruebas que se propone instalar en Guatemala.

Una parte importante a tomar en cuenta en este estudio es la descripción de los servicios con los que contará el Centro de Pruebas, dentro de esta parte se describen las pruebas y se clasifican de acuerdo a su tipo.

Para la realización de las pruebas, se necesitan herramientas, por esta razón, la siguiente parte consiste en la descripción de las herramientas que sirven de apoyo para la correcta ejecución de cada una de las pruebas.

Se presentan los resultados de una encuesta realizada a varios profesionales de la comunidad informática de Guatemala, acerca de las pruebas que se realizan dentro de las organizaciones a las que pertenecen y las herramientas que utilizan para la ejecución de las pruebas. Estos resultados nos darán un nivel de aplicación de las pruebas al software en Guatemala, lo cual será un indicativo valioso dentro del plan de implementación del Centro de Pruebas. Luego de esto se realiza un estudio de mercado para determinar la demanda potencial del centro de pruebas. En la siguiente parte, se define el

proceso productivo o prestación del servicio con las características tecnológicas y el equipo necesario.

Con base a lo anterior, se define el modelo de trabajo, la estructura organizacional y los objetivos de cada puesto.

Finalmente, se trata de mostrar la factibilidad del centro de pruebas en términos monetarios utilizando un flujo de efectivo del primero año de ejecución del centro de pruebas.

1. NATURALEZA DEL PROYECTO

1.1 Caso de estudio

1.1.1 Descripción general del Centro de ensayos de Software

A finales del año 2004 se creó en Uruguay el Centro de Ensayos de Software. Las autoridades del CES han mostrado el deseo de apoyar la iniciativa de crear en Guatemala el Centro de Pruebas de Software, razón por la que en este plan se tomará como un caso de estudio.

Centro de Ensayos de Software

El CES es un emprendimiento realizado en conjunto con la Cámara Uruguaya de Tecnologías de la Información (CUTI) y el Instituto de Computación (InCo) de la Facultad de Ingeniería de la Universidad de la República, a través de la Fundación Julio Ricaldoni.

El objetivo principal del CES es contribuir a la mejora de la competitividad y productividad de la industria de Tecnologías de la Información.

El CES tiene para su funcionamiento la siguiente estructura:

Un laboratorio de *Testing* para la evaluación de productos desde el punto de vista funcional.

Un laboratorio de Ensayos de Plataformas para realizar pruebas de desempeño y resolver problemas de funcionamiento en arquitecturas de hardware y software complejas.

También cuenta con un Observatorio Tecnológico, este tiene como objetivo difundir las novedades y tendencias en el área de Tecnologías de la Información y facilitar el acceso a las mismas por parte de la Industria.

La visión del CES dice así: “Ser un símbolo de calidad en software y garantía de resultado objetivo, basado en pruebas reales. Ser un referente para la industria de TI en tendencias, tecnologías y resultados.

La misión del CES es: *Brindar servicios para evaluar la calidad de los productos. Instrumentar prácticas para asegurar la rigurosidad en los servicios. Anticipar los cambios tecnológicos ofreciendo servicios e información sobre las tecnologías de última generación. Contribuir a la construcción de un motor de innovación en el área de la TI, promoviendo la participación en el desarrollo tecnológico.*

1.1.2 Servicios y productos

Los servicios que tiene el CES son básicamente tres:

Consultoría

Asesorar a una empresa en la mejora de los procesos de *testing*, definición de estrategias y automatización de las pruebas.

Colaboran en la creación y consolidación de las áreas de *testing*.

Capacitación

Realizar programas de capacitación en la disciplina de *testing* según las necesidades de la empresa.

Servicios de *Testing*

Realizar el *testing* en las aplicaciones de las empresas de manera efectiva y controlada, definiendo claramente el contexto y los objetivos.

Proponer una estrategia de *testing* basada en el análisis de riesgos del producto.

Test funcional: detecta discrepancias entre el funcionamiento real del producto y el deseado de acuerdo a su especificación funcional.

Test de integración: estudia el comportamiento de la integración de distintos módulos o componentes del producto.

Test de configuración: evalúa y analiza el comportamiento del producto en plataformas heterogéneas.

Test de robustez: estudia el comportamiento del producto en casos que no fueron previstos en su especificación.

Test de compatibilidad: estudia y analiza el comportamiento del producto cuando éste interactúa con otros sistemas.

Test de stress: estudia el rendimiento del producto sometido a los límites de carga para los cuales fue diseñado.

Test de carga: estudia el rendimiento del producto ante una carga para la cual fue diseñado.

Test de escalabilidad: estudia la evolución del rendimiento del producto frente a cambios en los recursos disponibles.

El CES dispone de profesionales especializados en técnicas, herramientas y metodologías de *testing*, con flexibilidad para adaptarse a diferentes ambientes de trabajo y rápida propuesta. Garantiza la confidencialidad de la información involucrada en los proyectos, brindando neutralidad e independencia al realizar las pruebas. Cuenta con una amplia infraestructura tecnológica a disposición de sus clientes. Realiza *testing* independiente, permitiendo mejorar la calidad del producto antes de su puesta en producción, evitando sorpresas no deseadas y aumentando la satisfacción del cliente.

La propuesta de trabajo del CES involucra aspectos académicos con las mejores prácticas surgidas de la experiencia en la industria y los contactos directos con instituciones regionales e internacionales que trabajan en la materia. El CES busca conocer las expectativas de sus clientes y recorrer juntos el camino de la mejora continua de la calidad del software nacional.

Resultados obtenidos

Como resultado de las pruebas realizadas al software, el Centro de Ensayos de Software emite un informe resumiendo las condiciones en las cuales se ejecutó al prueba y como se comportó el sistema.

1.2 Benchmarking

Durante el desarrollo de este trabajo de graduación tuve la oportunidad de participar en una videoconferencia organizada por la comisión de SOFEX dentro del programa “Preparando a Guatemala para exportar Servicios y Productos Tecnológicos”. El conferencista invitado fue Ing. Ignacio Barmak de la empresa Grupo Soluciones INNOVA de Argentina. A continuación se resumen los temas tratados:

Reseña

El centro de testeo de Argentina inició con poca inversión, un ancho de banda suficientemente alto, algunos servidores, y 4 personas en la ejecución de las pruebas. Hoy en día tienen la capacidad de realizar pruebas de rendimiento en varios servidores y pueden tener acceso a varios ambientes de pruebas.

Las pruebas pueden aplicarse a cualquier software, incluso los productos maduros. SAP recomienda realizar pruebas de software debido a los cambios en las configuraciones en las actualizaciones.

Construcción de un plan de pruebas

El plan de pruebas evoluciona conforme a las decisiones que se toman con respecto de la tecnología utilizada. Una prueba evalúa la calidad de los requerimientos. Se recomienda RUP ya que es la metodología más adecuada para realizar pruebas.

Certificación

GSI utiliza como sustento de su trabajo los siguientes modelos:

ISO 9002: Sistema de calidad-Modelo para el aseguramiento en calidad en producción e Instalación, estableciendo la prevención, detección y corrección de problemas durante la producción e instalación.

CMMI: El **CMMI** significa decir que se hace, hacer lo que se dice y medirlo [1].

Capability Maturity Model Integration (CMMI) [2] es un modelo para la mejora de procesos que proporciona a las organizaciones los elementos esenciales para procesos eficaces. Las mejores prácticas CMMI se publican en los documentos llamados modelos. En la actualidad hay dos áreas de interés cubiertas por los modelos de CMMI: Desarrollo y adquisición.

MoProSoft [3]: **Modelo de Procesos para la Industria del Software.** Modelo para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software. Desarrollado por la Asociación Mexicana para la Calidad en Ingeniería de Software [4] a través de la Facultad de Ciencias de la Universidad Nacional Autónoma de México (UNAM) y a solicitud de la Secretaría de Economía para obtener una norma

mexicana que resulte apropiada a las características de tamaño de la gran mayoría de empresas mexicanas de desarrollo y mantenimiento de software. Moprosoft es el nombre del modelo en la comunidad universitaria y profesional, y la norma técnica a la que da contenido es la NMX-059/01-NYCE-2005 que fue declarada Norma Mexicana el 15 de agosto de 2005, con la publicación de su declaratoria en el Diario de la Federación.

Moprosoft considera que los modelos de evaluación y mejora CMMI e ISO/IEC 15504 no resultan apropiados para empresas pequeñas y medianas de desarrollo y mantenimiento de software. Sobre las áreas de procesos de los niveles 2 y 3 del modelo SW-CMM e inspirándose en el marco de ISO/IEC 15504 se ha desarrollado este modelo.

Resultado de las pruebas

El resultado de las pruebas siempre es opinable. Las pruebas absolutas no existen. Una prueba al 100% no existe. Lo que se puede asegurar es que el producto va a funcionar en ciertas condiciones dadas apoyado por investigación y desarrollo permanente. El producto de software sale menos contaminado. Se escribe un informe con las pruebas realizadas y los defectos encontrados, alineados con el plan de pruebas.

Beneficios y Costos

Las pruebas ejecutándolas bien disminuyen el costo del proyecto. Un modelo adecuado disminuye el tiempo de construcción y le da más tiempo de venta. Tangible: El costo de la mano de obra se disminuye. Intangible: El desprestigio obtenido por un mal producto.

Los costos de las pruebas se reducen en mayor porcentaje cuando se tiene un modelo integrado, es decir, las personas dedicadas a las pruebas realizan una prueba y si encuentran defectos no tienen que esperar a que sean compuestos para volver a probar sino que utilizan el tiempo muerto para realizar otra prueba a un producto diferente.

Valor agregado del centro de testeo

Adaptación del modelo CMMI. Para tener una certificación en CMMI se necesita un *Quality Assurance*, lo que implica tener las pruebas, estas pruebas pueden ser propias o contratadas de terceros.

El mejor paso para empezar según RUP es mitigar los peores riesgos primero. Se requiere de un trayecto largo para llegar a CMMI, la certificación CMMI es una acción que necesita un Fee bastante alto y periódico.

Mercado del Centro de Pruebas

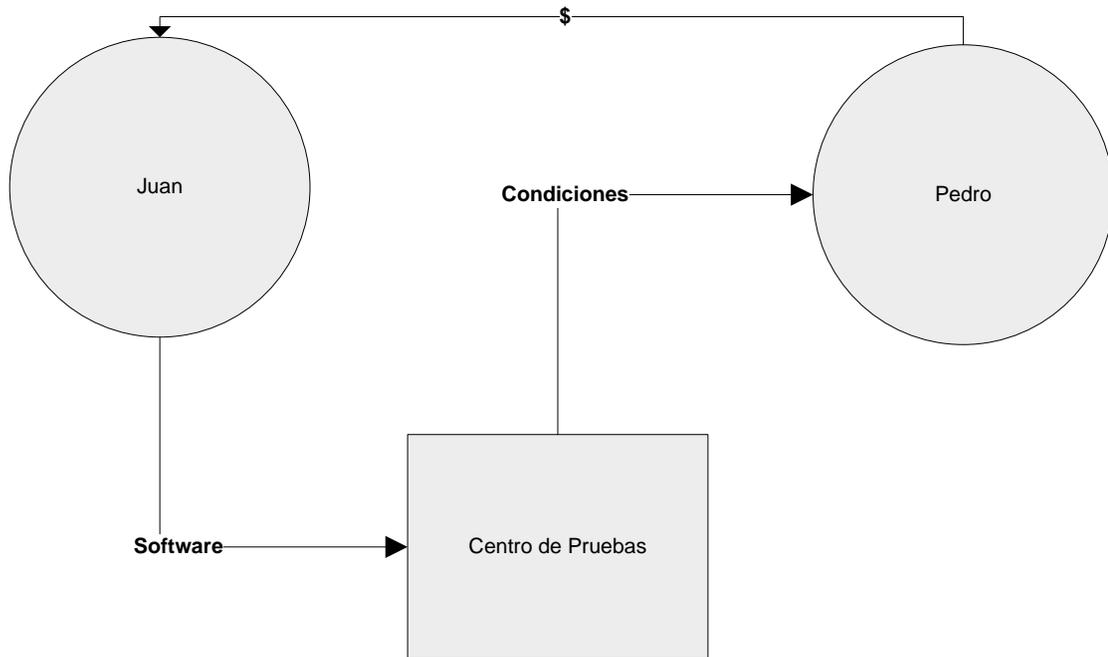
Los clientes del centro de pruebas de software son todos los que están ambicionando un proceso de calidad. GSI ha tenido la oportunidad de ver clientes que de comprar servicios de *testing* se pasaron a tener su propio centro de pruebas. Para tener un mercado mayor se necesita realizar un trabajo de concientización. Motivar a los empresarios y formar gente capacitada para realizar las pruebas.

Capacitación

Un tester junior tiene un plan de capacitación de 5 a 6 meses. El perfil de un tester junior debe de ser alguien concentrado y muy creativo. Una persona a mitad de la carrera de informática.

Modelo del cliente final

Figura 1. Modelo comercial



El modelo comercial del centro de pruebas de software es que el productor de software le da el producto al centro y luego de las pruebas el centro le dice al consumidor las condiciones en las que se entrega el producto para que luego el consumidor le pague al productor por un producto funcional.

Madurez del mercado

Demanda se ha dado desde hace más de 3 años. La demanda sostenida de las pruebas se ha logrado desde hace un poco más de un año.

Centro en Guatemala

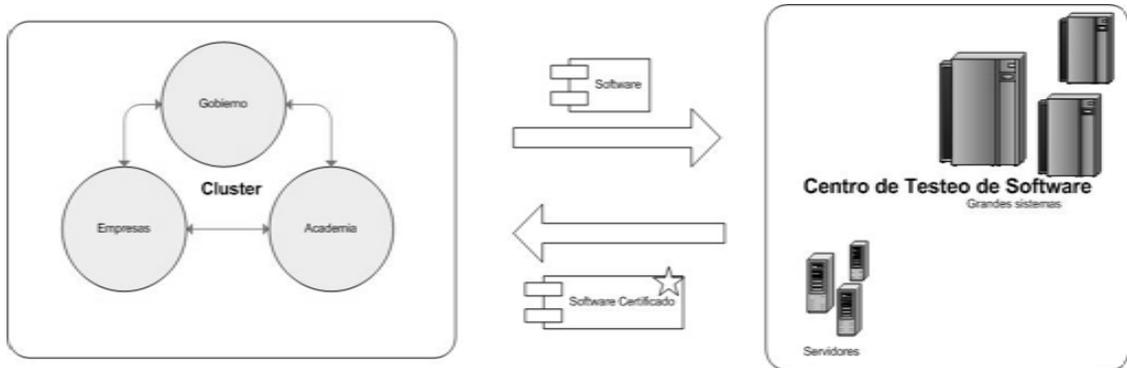
El Ing. Barmak hace algunas observaciones importantes para poder iniciar con el centro de pruebas de software en Guatemala.

- El centro de testeo al inicio se centrará en construir y ejecutar las pruebas manualmente, es decir que no incluiría pruebas robotizadas.
- No se pretende tener un zoológico de hardware para todos los tipos de prueba, más bien se requerirá acceso a los servidores para realizar las pruebas. Esto implica tener un enlace con suficiente ancho de banda, mínimo recomendado 3 Mbps.
- GSI ofrece apoyo a la formación del centro de pruebas en Guatemala y sugiere un modelo integrado con la Academia.
- El modelo de inversión debería de considerar a alguna autoridad que se interese por el proyecto.

1.3 Descripción del Centro de Pruebas (Guatemala)

El 15 de junio de 2005 quedó formada la Comisión de Software de Guatemala, SOFEX, como un sector organizado dentro de AGEXPORT (Asociación de Gremiales de Exportadores de Productos No Tradicionales), con la visión de ser la organización que impulse la industria del desarrollo de software, para ser la más creciente de Guatemala. Uno de los proyectos que apoyan esta visión es el centro de pruebas de software. Este centro tiene la función de certificar todo software desarrollado en el país y en otros países.

Figura 2. Rol del Centro de Pruebas de Software



Una vez que el software esté certificado, éste puede ser exportado con la garantía de que es un software de calidad.

Esto beneficiará tanto a las empresas que compran software como a las que lo producen generando una mayor confianza en los productos y servicios.

Todo gran proyecto es exitoso si se planea correctamente, es por eso que el plan del proyecto es de gran importancia para todos los involucrados, inversionistas, directores y beneficiarios.

El trabajo de graduación consistirá en la evaluación de la factibilidad de implantar este centro en nuestro país.

El negocio del Centro de Pruebas es el control de calidad en sistemas de software. La función del centro es certificar a todo software con buena calidad.

El Centro de Pruebas toma como entrada a un sistema de software y los requerimientos bajo los cuales fue construido, el proceso consiste en realizarle pruebas al sistema y en base a los resultados de las pruebas se tiene como resultado el nivel de calidad del sistema.

El valor agregado del centro hacia los clientes es que los educa acerca de la importancia de la calidad y sobre como regular el proceso de determinar los estándares de calidad.

Misión:

Brindar servicios para evaluar la calidad de los productos. Avalar que el producto cumple con las especificaciones, bajo las cuales fue construido. Fomentar prácticas que aseguren la rigurosidad en los servicios. Participar en el desarrollo tecnológico del país elevando la calidad de los productos.

Visión:

Ser un distintivo de calidad en software. Proporcionar confianza absoluta al comprador y seguridad al productor.

Metas y objetivos del centro:

- Convertir un producto en un producto certificado, el cual es un producto que cumple a cabalidad con sus especificaciones.
- Garantizar un resultado objetivo, basado en pruebas reales.
- Clientes confiados en las pruebas de calidad de software.
- Ser líderes de Centroamérica en pruebas de software.

Filosofía del Centro de Pruebas:

La filosofía de la empresa es lo más importante para los empleados; pues teniendo todos la misma, se conseguirá que todos trabajen por ella y a la vez que se cumpla.

La filosofía de la empresa se resume en dos puntos:

- Mantener la calidad de los productos de software.
- Elevar el nivel de calidad para permitir a las empresas guatemaltecas competir a nivel internacional y obtener reconocimiento.

A quién beneficia el centro:

Se benefician las empresas de desarrollo de software, principalmente porque se garantiza la calidad de los productos.

Las empresas que utilizan software, el centro de pruebas les proporciona la seguridad de que los productos obtenidos cumplen a cabalidad todos sus requerimientos.

Industria de Pruebas de Software. Industria en crecimiento. Cambios que se pueden predecir en la industria a corto y largo plazo. Como el centro se va a acoplar para tomar ventaja de los cambios.

La industria de Pruebas de Software ha tomado gran importancia últimamente. Las pruebas a un sistema de software son de gran importancia por lo que se deben realizar objetivamente y sin ningún tipo de sesgo por parte de la empresa desarrolladora o del cliente. El Centro de Pruebas es un ente imparcial e intermedio que ayuda a descubrir errores que los desarrolladores no ven durante todo el ciclo de desarrollo y garantiza la correcta funcionalidad a los clientes.

Tener éxito en la industria de software toma más que solamente entregar un producto de calidad a tiempo y de una forma eficiente. Los productos de software de hoy en día están aumentando en su complejidad y casi siempre operan en varias capas sobre ambientes multiplataforma. Esos productos son creados en ambientes de desarrollo rápido que los llevan a la cabeza del mercado rápidamente. Las pruebas de software han sido adecuadas a un ciclo de desarrollo rápido cuando los requerimientos de ahora son más extensos que nunca.

Fortalezas y competencias:

- Es el primer Centro de Pruebas en Guatemala
- Se tiene el apoyo del CES en Uruguay
- Se tiene el apoyo de SOFEX
- Experiencia en desarrollo de software
- Profesionales especializados en control de calidad, capacitados según los estándares del ISTQB.
- Las empresas nacionales tienen que crear una política de control de calidad que incluya al Centro de Pruebas

1.4 Productos y servicios

Un grupo de actividades pueden apuntar a la verificación de un sistema de software, o parte del sistema, basadas en una razón específica u objetivo para las pruebas.

Un tipo de prueba se enfoca en un objetivo de la prueba en particular, el cual puede ser probar una función que debe ser realizada por el sistema; la

calidad de una característica no funcional, tal como la confiabilidad o la utilidad, la estructura o la arquitectura del software o del sistema; o relacionados a cambios, por ejemplo confirmando que los defectos han sido arreglados (pruebas de confirmación) y buscando cambios involuntarios (pruebas de regresión).

Un modelo del software puede ser desarrollado y/o usado en las pruebas estructurales y funcionales. Por ejemplo en las pruebas funcionales será de mucha utilidad un modelo del flujo del proceso, un modelo de los estados del sistema y sus transiciones, o las especificaciones del lenguaje utilizado; y para las pruebas estructurales un modelo de control de flujo o un modelo de la estructura del menú.

A continuación se presentan una clasificación de las pruebas, las cuales contienen las pruebas que el Centro de Pruebas proveerá.

1.4.1 Pruebas funcionales

Las funciones que un sistema, subsistema o componente deben realizar pueden estar descritas en la especificación de los requerimientos, casos de uso o especificaciones funcionales, o estas funciones pueden estar sin documentar. Las funciones son “lo que el sistema hace”.

Las pruebas funcionales se basan en esas funciones y características (descritas en documentos y comprendidas por los examinadores), y pueden ser ejecutadas en cualquier nivel de las pruebas

Test funcional

Detecta discrepancias entre el funcionamiento real del producto y el deseado de acuerdo a su especificación funcional.

1.4.2 Pruebas no funcionales

Las pruebas no funcionales incluyen pruebas de funcionamiento, pruebas de carga, pruebas de estrés, pruebas de usabilidad, pruebas de interoperabilidad, pruebas de capacidad de mantenimiento, pruebas de confiabilidad y pruebas de portabilidad. Las pruebas no funcionales no solo se limitan a las mencionadas anteriormente.

Son las pruebas de “como trabaja el sistema”.

El término “prueba no funcional” describe las pruebas requeridas para medir las características de los sistemas y el software que pueden ser cuantificadas en una escala variable, tales como tiempos de respuesta para las pruebas de funcionamiento. Estas pruebas pueden ser referidas a un modelo de calidad tal como alguno de los definidos en “*Software Engineering – Software Product Quality*” (ISO 9126).

Test de escalabilidad

Estudia la evolución del rendimiento del producto frente a cambios en los recursos disponibles.

Test de configuración

Evalúa y analiza el comportamiento del producto en plataformas heterogéneas.

Test de robustez

Estudia el comportamiento del producto en casos que no fueron previstos en su especificación.

Test de carga

Estudia el rendimiento del producto ante una carga para la cual fue diseñado.

Test de estrés

Estudia el rendimiento del producto sometido a los límites de carga para los cuales fue diseñado.

Test de usabilidad

La usabilidad la confirman una serie de métricas y métodos que buscan hacer que un sistema sea fácil de usar y de aprender. Al hablar de sistema la referencia se hace a cualquier dispositivo que tenga que ser operado por un usuario. En esta categoría caen los sitios Web, aplicaciones de software, hardware, etc.

Estudios recientes en el área de interfaces humano–computadora indican que los *tests* de usabilidad muestran dónde están los problemas.

Un *test* de usabilidad es una medida empírica de la usabilidad de una herramienta, sitio o aplicación, tomada a partir de la observación sistemática de usuarios llevando a cabo tareas reales.

Para llevar a cabo un *test*, debe haber por lo menos 3 roles: el participante o usuario, el facilitador o monitor y los observadores. Sin embargo, en el enfoque llamado *Discount usability engineering* se plantea una manera mucho más simple y menos costosa de hacer *tests*, basándose en la regla 80/20: el 80% de los conocimientos y hallazgos de un *test* completo con el 20% de la inversión.

Las principales etapas de un *test* de usabilidad son: planificación, *test* y conclusión.

Test de accesibilidad

La accesibilidad indica la facilidad con la que algo puede ser usado, visitado o accedido en general por todas las personas, especialmente por aquellas que poseen algún tipo de discapacidad.

En informática, la accesibilidad incluye ayudas como las tipografías de alto contraste o gran tamaño, magnificadores de pantalla, lectores y revisores de pantalla, programas de reconocimiento de voz, teclados adaptados, y otros dispositivos apuntadores y de entrada de información.

La accesibilidad aplicada al contenido de Internet se denomina accesibilidad Web. En la Web, el W3C ha desarrollado directrices o pautas específicas para permitir y asegurar este tipo de accesibilidad. El grupo de trabajo dentro del W3C encargado de promoverla es el WAI (Web Accessibility Initiative).

1.4.3 Pruebas estructurales

Las técnicas estructurales son mejor utilizadas luego de las técnicas basadas en las especificaciones, para ayudar a medir la minuciosidad de las pruebas con el gravamen de la cobertura de un tipo de estructura.

La cobertura es el grado que una estructura ha sido ejercitada por una serie de pruebas, expresado en un porcentaje de los ítems que han sido cubiertos. Si la cobertura no es el 100% entonces se deberán de diseñar mas pruebas para cubrir los ítems que quedaron pendientes y de allí incrementar la cobertura.

Estas pruebas se realizan especialmente en los componentes y en las pruebas de integración de los componentes, las herramientas pueden ser utilizadas para medir la cobertura del código de los elementos, tales como sentencias y decisiones. Las pruebas estructurales pueden ser basadas en la arquitectura del sistema, tal como la herencia.

Test de compatibilidad

Estudia y analiza el comportamiento del producto cuando éste interactúa con otros sistemas.

Test de integración

Estudio del comportamiento de la integración de distintos módulos o componentes del producto.

Pruebas de confirmación y regresión

Cuando un defecto es detectado y arreglado el software debe ser probado de nuevo para corroborar que el defecto ha sido removido por completo. Esto se llama pruebas de confirmación. Debugging (arreglar defectos) es una actividad del desarrollo y no se considera una actividad de las pruebas.

Las pruebas de regresión son las repetidas pruebas de un programa ya probado, después de realizar un cambio para descubrir cualquier defecto introducido o sin descubrir como resultado del cambio realizado. Estos defectos pueden estar tanto en el software que está siendo probado como en alguno de los componentes del software. Estas pruebas de regresión son aplicadas cuando el software o el ambiente han cambiado. El grado de las pruebas de regresión es basado en el riesgo de no encontrar defectos en el software que anteriormente trabajaba bien.

Las pruebas deben ser repetitivas si se van a usar para pruebas de confirmación o para apoyar las pruebas de regresión.

Las pruebas de regresión aplican a las pruebas funcionales, no funcionales y estructurales. Los paquetes de pruebas de regresión se ejecutan

muchas veces y evolucionan lentamente, por lo que las pruebas de regresión son buen candidato para la automatización.

1.5 Herramientas de apoyo

Existen varias herramientas que apoyan diferentes aspectos de las pruebas. Se pueden clasificar herramientas de acuerdo a las actividades de las pruebas a las que apoyan.

Algunas herramientas apoyan únicamente una actividad, algunas otras pueden apoyar más de una, pero su clasificación va de acuerdo a la actividad a la cual apoya más. Algunas herramientas comerciales ofrecen apoyo para únicamente una actividad, algunas empresas ofrecen paquetes de herramientas o una gama de herramientas que apoyan todas estas actividades.

Las herramientas pueden ayudar a mejorar la eficiencia de las actividades automatizando tareas repetitivas. También pueden ayudar a mejorar la confiabilidad de las pruebas, por ejemplo, automatizando comparaciones de una gran cantidad de datos o simulando el comportamiento del software.

Algunas herramientas pueden ser desequilibrantes, esto quiere decir que la herramienta en si afecta el resultado de las pruebas. Por ejemplo, el tiempo de respuesta actual puede ser diferente dependiendo de cómo se mida con una herramienta de funcionamiento diferente. La consecuencia que tienen estas herramientas desequilibrantes se conoce como el efecto de la punta de prueba, o en inglés "*probe effect*".

Con la finalidad que el Centro de Pruebas certifique la calidad de los productos, es necesaria la utilización de las herramientas de apoyo correspondientes. A continuación se presentan las herramientas que apoyan las actividades realizadas por el Centro.

1.5.1 Herramientas de soporte para la administración de las pruebas y el test

Las herramientas de administración de las pruebas se utilizan para todas las actividades de pruebas en todo el ciclo de vida del software. Apoyan la creación, administración y control de documentos como los planes de las pruebas, especificaciones y resultados. Algunas herramientas apoyan aspectos de la administración del proyecto, como por ejemplo la planificación cronológica de las pruebas, el almacenamiento de los resultados y registrar problemas que son descubiertos durante las pruebas.

Estas herramientas almacenan los casos de prueba y su cobertura y sus enlaces con otros objetos e información.

1.5.1.1 Herramientas para la administración de las pruebas

Las características de estas herramientas incluyen:

- Apoyo a la administración de las pruebas y las actividades que se llevan a cabo.

- Interfaces para las herramientas de ejecución de las pruebas, herramientas de rastreo de los defectos y las herramientas de administración de los requerimientos.
- Control independiente de las versiones o interfaces un herramientas externas de administración de la configuración.
- Apoyo al seguimiento de las pruebas, a los resultados de las pruebas y a los incidentes basándose en documentos tales como los de especificación de requerimientos.
- Lleva una bitácora del resultado de las pruebas y genera reportes del progreso de las pruebas.
- Análisis cuantitativo (métricas) relacionado a las pruebas, por ejemplo la cantidad de pruebas fallidas y exitosas, y relacionado al objeto probado, como por ejemplo la cantidad de incidentes que generó el objeto. Todo esto con el fin de dar información sobre el objeto probado, y controlar y mejorar el proceso de las pruebas.

1.5.1.2 Herramientas para la administración de los requerimientos

Estas herramientas guardan las declaraciones de los requerimientos, realizan chequeos en busca de inconsistencias o requerimientos sin definir, permite priorizar los requerimientos y permite que las pruebas individuales puedan ser trazadas de acuerdo a los requerimientos, funciones y/o características. El seguimiento puede ser mostrado en los reportes del progreso

de la administración de las pruebas. La cobertura de los requerimientos, funciones y/o características abarcadas por una serie de pruebas también debe ser reportada.

1.5.1.3 Herramientas para la administración de los incidentes

Estas herramientas almacenan y administran los incidentes reportados, estos incidentes pueden ser defectos, fallas o problemas percibidos y anomalías. Apoyan la administración de los incidentes reportados de diferentes formas:

- Ayuda a priorizar los incidentes.
- Asigna acciones a las personas
- Atribuye estados a los incidentes

Estas herramientas habilitan el monitoreo de los incidentes en cualquier momento, a menudo proveen análisis estadístico y reportes de los incidentes. Estas herramientas son conocidas también como herramientas de seguimiento de defectos.

1.5.1.4 Herramientas para la administración de la configuración

Estas no son estrictamente herramientas para pruebas de software, pero son necesarias para llevar el control de las versiones y estructuras del software así como de las pruebas.

Las características de estas herramientas:

- Almacenan información acerca de las versiones y estructuras del software y el producto a probar.
- Habilitan el seguimiento entre el producto a probar y el producto probado con todas sus variantes.

Son de mucha utilidad durante el desarrollo sobre más de una configuración del ambiente del software o hardware, por ejemplo para diferentes versiones del sistema operativo, para diferentes compiladores, para diferentes navegadores o para diferentes computadoras.

1.5.2 Herramientas de soporte para las pruebas estáticas

1.5.2.1 Herramientas de apoyo para el proceso de revisión

Estas herramientas almacenan información acerca del proceso revisión, almacena y comunica los comentarios de las revisiones, reportes de los defectos y esfuerzos, administra las referencias para la revisión de las reglas y/o listas de comprobación (*checklists*) y mantiene la capacidad de rastreo entre los documentos y el código fuente. Estas herramientas también proveen ayuda para las revisiones en línea (*on-line*), lo cual es de mucha ayuda si el equipo está disperso geográficamente.

1.5.2.2 Herramientas para análisis estático

Estas herramientas son de mucha ayuda para los desarrolladores, los ejecutores de las pruebas y para el personal encargado del aseguramiento de la calidad en encontrar defectos antes de las pruebas dinámicas. El propósito de las pruebas incluye:

- La aplicación de estándares de codificación.
- El análisis de las estructuras y dependencias, por ejemplo los links hacia las páginas Web.
- Ayuda al entendimiento del código.

Las herramientas de análisis estático pueden calcular métricas del código, las cuales pueden proporcionar información valiosa, por ejemplo para planeación o para el análisis de riesgos.

Estas herramientas proveen información acerca de la calidad del código de un programa. Esto se consigue analizando el código del programa y no ejecutando los casos de prueba. Este tipo de herramientas habilitan una medición objetiva de varias características del software, por ejemplo la complejidad y otros atributos de calidad. Muchas de estas herramientas de análisis estático soportan de conformidad las reglas de programación convencionales. Cualquier infracción será resaltada con una advertencia de violación de estas reglas dentro del código. Estas herramientas vienen con un paquete de reglas estándares para cada lenguaje de programación y pueden ser adaptadas a los requerimientos específicos de un proyecto.

1.5.2.3 Herramientas de modelado

Estas herramientas son capaces de validar los modelos del software. Por ejemplo, un verificador de modelos de base de datos puede encontrar defectos e inconsistencias en un modelo de datos; otras herramientas de modelado pueden encontrar defectos en un modelo de estados o en un modelo de objetos. Estas herramientas a menudo pueden ayudar en la generación de algunos casos de prueba basados en el modelo.

El mayor beneficio de una herramienta de análisis estático y de una herramienta de modelado es el costo de la eficacia en encontrar más defectos en el menor tiempo durante el proceso de desarrollo. Como resultado, el proceso de desarrollo se puede acelerar y mejorar teniendo menos trabajo repetitivo.

1.5.3 Herramientas de soporte para la especificación de las pruebas

1.5.3.1 Herramientas de diseño de pruebas

Estas herramientas generan entradas para las pruebas o las pruebas en sí para los requerimientos, para las interfaces gráficas de usuario, para los modelos de diseño (estado, datos u objetos), o para el código. Este tipo de herramientas puede generar los resultados tal como fueron previstos. Las pruebas generadas de un modelo de estado o de un modelo de objetos son de utilidad para verificar la implementación del modelo en el software, pero son raramente suficientes para verificar todos los aspectos del sistema o del

software. Estas herramientas pueden ayudar a ahorrar tiempo valioso e incrementan la minuciosidad de las pruebas por lo completo de las pruebas que la herramienta puede generar.

Otras herramientas dentro de esta categoría ayudan en la generación de pruebas proveyendo plantillas estructuradas, llamadas marcos de pruebas, que generan las pruebas o trozos de las pruebas y así acelerar el proceso de diseño de las pruebas.

Las herramientas para diseñar entradas para las pruebas generan estas entradas de las especificaciones que normalmente las obtienen de herramientas CASE en forma de requerimientos formalmente especificados.

También existen herramientas que pueden generar entradas para las pruebas basadas en el análisis del código de un programa. Estas herramientas también pueden ejecutar los casos de prueba, como por ejemplo cuando se quieren probar las fallas en el manejo de tipos de datos incompatibles.

1.5.3.2 Herramientas de preparación de datos de prueba

Estas herramientas manipulan las bases de datos, archivos o las transferencias de datos para instalar datos de prueba que serán utilizados durante la ejecución de las pruebas. El beneficio de estas herramientas es asegurar que la transferencia de los datos hacia el ambiente de pruebas se realiza de una forma anónima para la protección de los datos.

1.5.4 Herramientas de soporte para la ejecución de las pruebas y el log

1.5.4.1 Herramientas para la ejecución de pruebas

Estas herramientas permiten que las pruebas se realicen automáticamente o semi-automática, utilizando entradas predeterminadas y resultados esperados a través del uso de un lenguaje de scripts. Este lenguaje de scripts hace posible la manipulación de las pruebas con poco esfuerzo, por ejemplo, para repetir la prueba con diferentes datos o para probar otra parte del sistema con los mismos pasos. Generalmente estas herramientas incluyen características de comparaciones dinámicas y proveen una bitácora de cada una de las pruebas ejecutadas.

Este tipo de herramientas también pueden ser utilizadas para grabar todas las pruebas cuando se configuran para capturar las actividades. Capturar las entradas de las pruebas durante un *test* exploratorio o ejecutar pruebas sin especificar pueden resultar de mucha ayuda para reproducir o documentar la prueba en caso de que se encuentre un defecto.

Existen robots para la ejecución de las pruebas, se concentran en *capturar y reproducir*, lo que hacen estos robots es mostrar las entradas del usuario y las respuestas del sistema en una pantalla para que estas puedan ser comparadas después. Para aplicaciones que poseen interfaces graficas de usuario (GUI) las herramientas pueden simular movimientos del ratón, teclas presionadas y reconocer objetos gráficos como las ventanas, diálogos, campos y otros elementos de control. Las condiciones de los objetos gráficos pueden ser almacenadas para ser comparadas después.

Las secuencias de ejecución son normalmente almacenadas en scripts programables, que son mayormente modificados manualmente para introducir los pasos de verificación y mejorar el mantenimiento del *script*. Uno de los aspectos más importantes a tomar en cuenta es a menudo la separación de los scripts de prueba de los datos de prueba. Las entradas del usuario se repiten durante la ejecución del *script* y las respuestas del sistema son comparadas con las que se encuentran almacenadas en el *script*. Las diferencias entre las respuestas esperadas y las obtenidas son identificadas por la herramienta y algunas veces automáticamente se crea un registro en una herramienta de administración de incidentes. Los casos de prueba, los datos de prueba, scripts y resultados esperados pueden ser almacenados en diferentes archivos de prueba. Los robots para la ejecución de pruebas son usualmente utilizados para la automatización de las pruebas de regresión.

1.5.4.2 Herramientas para el marco de pruebas unitarias y pruebas en conjunto

Las pruebas en conjunto pueden facilitar las pruebas de los componentes o una parte del sistema simulando el ambiente dentro del cual el objeto a probar se ejecutará. Esto se puede realizar porque otros componentes del ambiente todavía no están disponibles y son reemplazados por partes y/o drivers, o simplemente para poder proveer un ambiente predecible y controlable dentro del cual cualquier falla sobre el objeto puede ser localizada.

Un marco puede ser creado donde una parte del código, objeto, método o función, unidad o componente puede ser ejecutado, llamando al objeto que va a ser probado y/o proporcionándole retroalimentación a ese objeto. Esto se puede

realizar proporcionando medios artificiales de proveer entradas al objeto de prueba y/o proporcionando partes de las salidas del objeto en lugar de las verdaderas.

Las herramientas de pruebas en conjunto pueden ser utilizadas también para proveer un marco de ejecución dentro de un middleware, en donde los lenguajes, sistemas operativos o hardware deben de ser probados todos juntos.

Estas herramientas pueden ser llamadas herramientas para un marco de pruebas unitarias cuando tienen un objetivo en particular en el nivel de prueba del componente. Este tipo de herramientas ayudan a la ejecución de las pruebas de los componentes de forma paralela con la construcción del código.

Los marcos de prueba y los drivers son utilizados para ejecutar software que no tiene interfaz de usuario o en donde la interfaz de usuario no se encuentra disponible.

1.5.4.3 Comparadores de pruebas

Las pruebas de comparación determinan las diferencias entre archivos, bases de datos e incluso entre resultados de pruebas. Las herramientas de ejecución de pruebas normalmente incluyen comparadores dinámicos, pero las comparaciones luego de las pruebas pueden ejecutarse por una herramienta de comparación. Un comparador de pruebas puede utilizar un oráculo, especialmente si las pruebas están automatizadas.

Las diferencias entre los resultados esperados y los obtenidos se almacenan utilizando herramientas de comparación. Estas herramientas de

comparación normalmente trabajan con una variedad de formatos de archivos bases de datos diferentes. A menudo son parte de los robots para las pruebas y pueden manejar interfaces de usuario orientadas a caracteres y/u objetos gráficos.

Las pruebas a los oráculos proporcionan los resultados esperados. Una prueba al oráculo es por ejemplo, un sistema antiguo que está siendo reemplazado por uno nuevo con la misma funcionalidad. El sistema antiguo proporciona los resultados esperados correctos para todos los casos de prueba. Los oráculos pueden ser utilizados por sistemas que deben de proporcionar mayor rendimiento. Un oráculo con pocas características de funcionamiento puede ser utilizado o creado para generar resultados funcionales que un sistema de alto rendimiento debe de proporcionar.

1.5.4.4 Herramientas de medición de la cobertura

Este tipo de herramientas pueden ser desequilibrantes o no-desequilibrantes, dependiendo de las técnicas de medición utilizadas, de lo que es medido y del lenguaje de programación. Estas herramientas miden el porcentaje de tipos específicos de estructuras de código que han sido probadas (por ejemplo, sentencias, decisiones y llamadas a módulos o funciones). Estas herramientas muestran minuciosamente como la medición del tipo de estructura ha sido realizada por un conjunto de pruebas.

Estas herramientas siguen la información detallada y la almacenan en archivos. Estas actividades hacen que el sistema se ponga lento y puede influir en la funcionalidad. Después de la ejecución de las pruebas, los datos

almacenados son analizados por la herramienta para generar estadísticas de la cobertura de las pruebas.

1.5.4.5 Herramientas de seguridad

Estas herramientas chequean que las computadoras estén libres de virus y de ataques de denegación de servicios. Un *firewall* por ejemplo no es estrictamente una herramienta para realizar pruebas, pero se utiliza en las pruebas de seguridad. Algunas otras herramientas de seguridad tensan al sistema en busca de vulnerabilidades específicas.

Estas herramientas son mayormente utilizadas para aplicaciones de *E-commerce* y *E-business* o bien para sitios Web. Esta herramienta revisa todos los aspectos de la aplicación Web que están en peligro de uso erróneo a causa de un acceso no autorizado.

1.5.5 Herramientas de soporte para desempeño y monitoreo

1.5.5.1 Herramientas de análisis dinámico

Estas herramientas ayudan a encontrar defectos que son evidentes únicamente cuando el software está en ejecución, tales como dependencias del tiempo o escapes de memoria. Estas herramientas se utilizan típicamente en las pruebas de componentes e integración de componentes, y cuando se está probando un middleware.

Estas herramientas son mayormente utilizadas para encontrar las posibles causas de los siguientes errores: punteros sin asignar, aritmética incorrecta, asignaciones incorrectas, uso y liberación de la memoria, espacios en memoria y otros errores que no se pueden detectar durante el análisis estático.

1.5.5.2 Herramientas para pruebas de funcionamiento

Estas herramientas monitorean y reportan el comportamiento del sistema sobre una variedad de condiciones de uso simuladas. Estas herramientas simulan una carga al sistema, a una base de datos, o a un ambiente del sistema, tal como una red o un servidor. Estas herramientas son nombradas a menudo por el aspecto del funcionamiento que miden, como de carga o de estrés, por eso son conocidas también como herramientas para pruebas de carga o herramientas para pruebas de estrés. Son normalmente basadas en ejecuciones repetitivas automatizadas de las pruebas, controladas por parámetros.

Estas herramientas tienen dos tareas principales: la generación de carga y la medición de las transacciones de prueba.

Los generadores de carga pueden simular múltiples usuarios o grandes volúmenes de datos. La carga será generada por un driver, el cual simula la carga con varios usuarios simultáneos. Se realizan diferentes mediciones de las transacciones realizadas, las cuales luego son utilizadas para analizar los tiempos de respuesta. Estas herramientas proporcionan varios reportes y gráficas mostrando la relación entre los tiempos de respuesta y la carga.

1.5.5.3 Herramientas de monitoreo

Estas herramientas no son precisamente herramientas de pruebas pero proveen información valiosa para los propósitos de las pruebas y que no está disponible por otros medios.

Estas herramientas analizan continuamente, verifican y reportan el uso de recursos específicos del sistema, y alerta de posibles problemas con los servicios. Almacenan información acerca de la versión y la estructura del software, y permite el rastreo de esta información.

Estas herramientas son mayormente utilizadas sobre aplicaciones de *E-commerce* o *E-business* y chequean si un Sitio Web proporciona la disponibilidad y el funcionamiento requerido por el usuario. Estas herramientas se ejecutan permanentemente en *background* y alertan cuando un sistema ya no está disponible o algún límite de funcionamiento ha sido violado.

1.6 Nivel de pruebas en Guatemala

Durante el desarrollo de este trabajo se realiza un estudio de la aplicación de las pruebas de software en Guatemala.

Las dos razones principales por las que realizamos pruebas son: para hacer un juicio acerca de la calidad y aceptabilidad, y para descubrir problemas. Probamos porque sabemos que errar es de humanos, esto es especialmente cierto en el ambiente del software y los sistemas controlados por software.

La herramienta utilizada para este estudio fue una encuesta, la cual tiene como objetivo principal, definir el nivel de las pruebas de software en Guatemala.

Esta encuesta fue contestada por 162 personas en su mayoría egresados de la Escuela de Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala.

La primera parte de la encuesta realizada obtiene información acerca de la percepción de las pruebas por parte del encuestado.

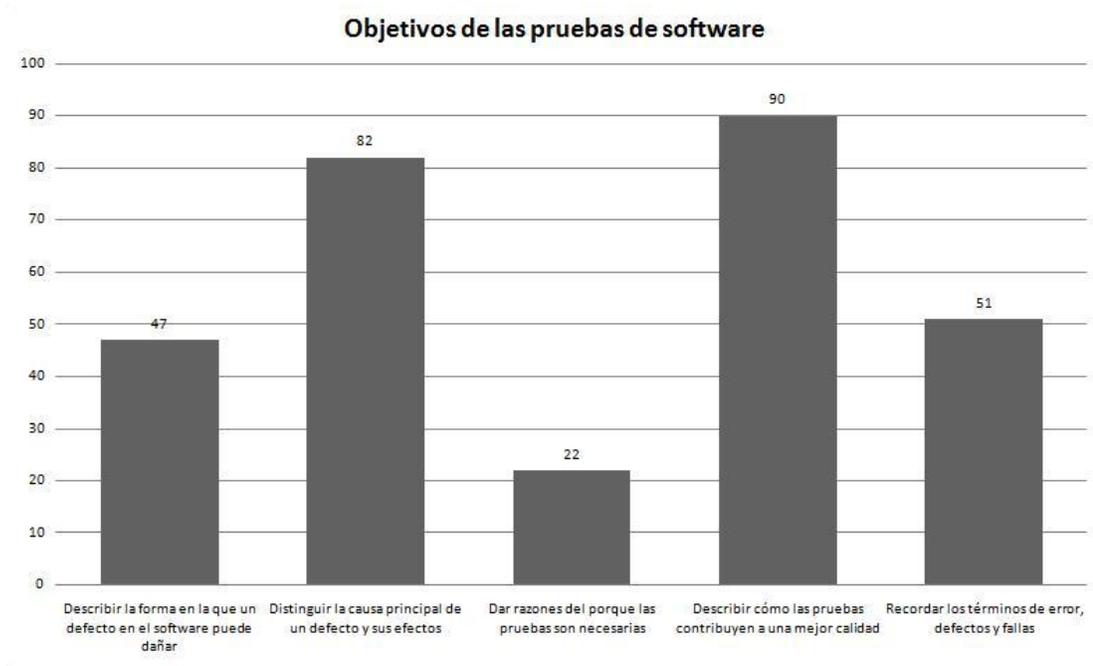
1.6.1 Generales

Dentro de la organización se realizan pruebas al software

1.6.1.1 Las pruebas al software tienen como principales objetivos los siguientes:

1	Describir con ejemplos, la forma en que un defecto en el software puede dañar a una persona, al ambiente o a una compañía.
2	Distinguir la causa principal de un defecto y sus efectos.
3	Dar razones del porqué las pruebas son necesarias dando ejemplos.
4	Describir porqué las pruebas forman parte de la aseguración de la calidad y dar ejemplos de cómo las pruebas contribuye a una mejor calidad.
5	Recordar los términos de error, defectos, fallas y términos correspondientes de error y <i>bugs</i> .

Figura 3. Objetivo de las pruebas de software



El equipo encargado del aseguramiento de la calidad es el ejecutor de las pruebas a todo sistema de software.

Principalmente se realizan estas pruebas para poder distinguir la causa principal de un defecto en el sistema y los efectos que este pueda causar si no es corregido. Al mismo nivel de importancia se encuentra que estas pruebas son realizadas para poder describir por qué las pruebas son importantes para el aseguramiento de la calidad y de cómo estas contribuyen a que el sistema tenga una mejor calidad.

Un porcentaje menor aplica las pruebas para poder recordar los términos de error, defectos o fallas que el sistema pueda tener. Una base de datos de conocimientos se va alimentando cada vez que las pruebas son realizadas y así

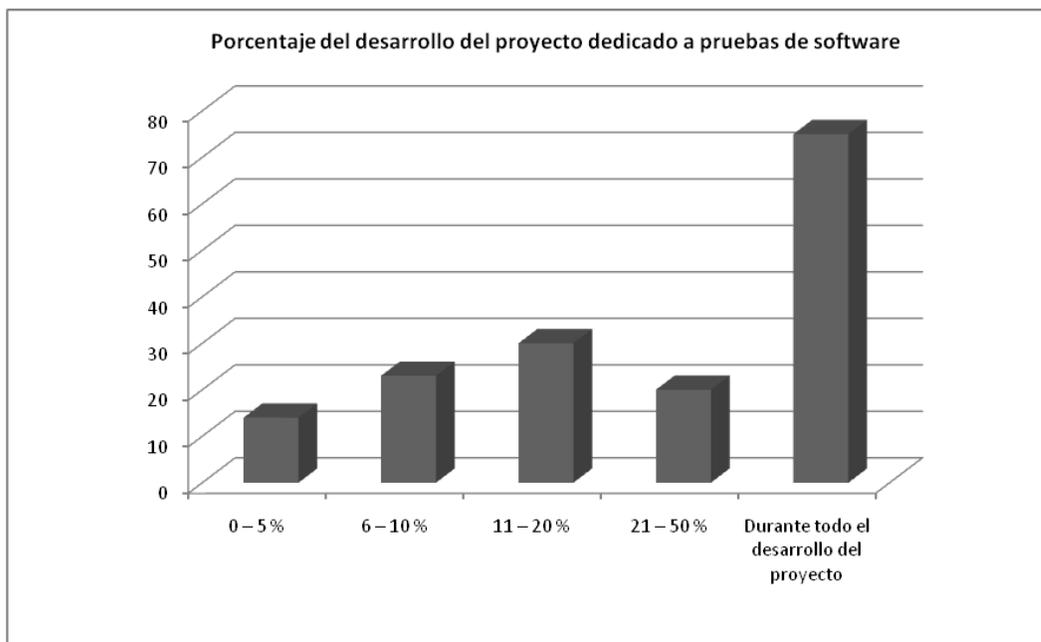
las correcciones a los errores encontrados serán más fáciles cada vez que aparezcan.

Al equipo ejecutor de las pruebas le interesa describir la forma en la que un defecto puede dañar a una persona, al ambiente o a una compañía. El equipo no busca justificar que las pruebas son necesarias al encontrar errores.

1.6.1.2 Porcentaje del desarrollo del proyecto dedicado a pruebas de software

El aseguramiento de la calidad del software es proporcional a las pruebas realizadas al software.

Figura 4. Porcentaje dedicado a pruebas de software



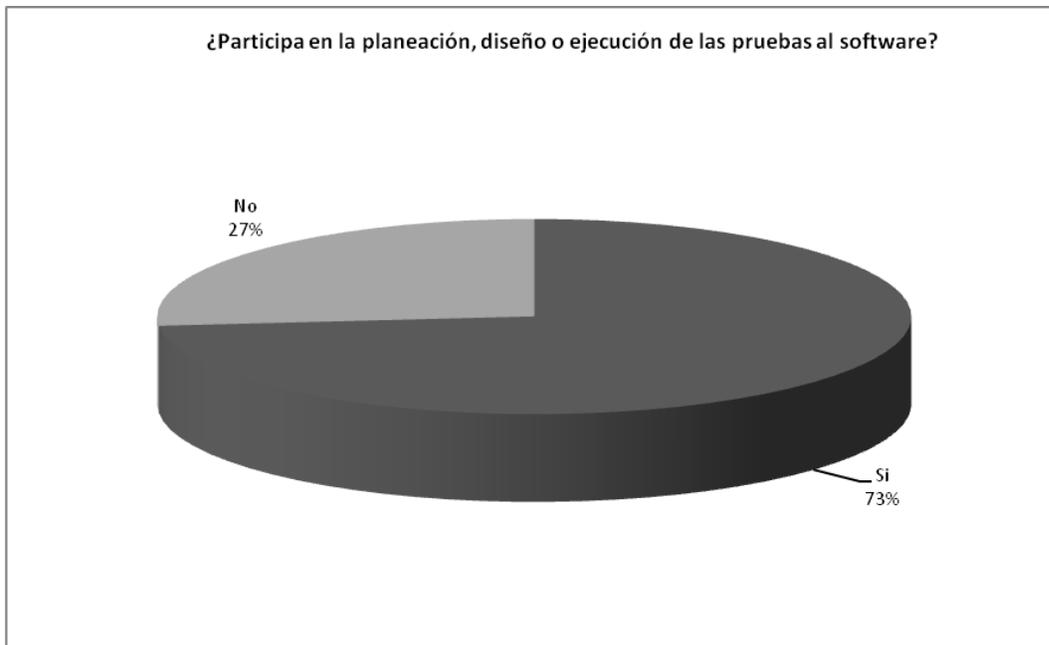
El diseño, planificación y ejecución de las pruebas debe de incluirse dentro del proyecto, queda claro que el tiempo dedicado a las pruebas de software principalmente se da durante todo el desarrollo. Esto puede deberse a que la metodología de desarrollo utilizada permita realizar dichas pruebas durante el desarrollo.

En segundo lugar, el tiempo dedicado a probar el software es entre el 11% y el 20% del desarrollo del proyecto. En tercer lugar se encuentra que entre el 6% y el 11% del desarrollo del proyecto se lo dedican a pruebas del software. En cuarto lugar tenemos que entre el 21% y el 50% del tiempo de desarrollo del proyecto es dedicado a pruebas.

Una menor parte solamente le dedica menos del 5% del tiempo a las pruebas

1.6.1.3 Participación en la planeación, diseño o ejecución de las pruebas al software

Figura 5. Participación en las pruebas de software



Se puede observar que la mayoría participa en alguna de las fases de las pruebas efectuadas al software, por lo que si existe participación por parte del personal para el aseguramiento de la calidad del software.

En este aspecto el equipo encargado del aseguramiento de la calidad está conformado por un número aceptable de colaboradores, el 73% tiene una participación en el equipo dejando fuera únicamente al 27%.

Esta participación tan grande se tiene relación con que las pruebas del software se realizan durante todo el desarrollo del proyecto por lo que en cada etapa del desarrollo debe de existir un grupo que se encargue del aseguramiento de la calidad para continuar con la siguiente etapa.

De igual forma cuando se desarrolla por módulos cada equipo desarrollador del módulo debe de tener cierto número de personas que aplican las pruebas al software, en la integración de los módulos también existe una parte de equipo dedicada a las pruebas finales de integración.

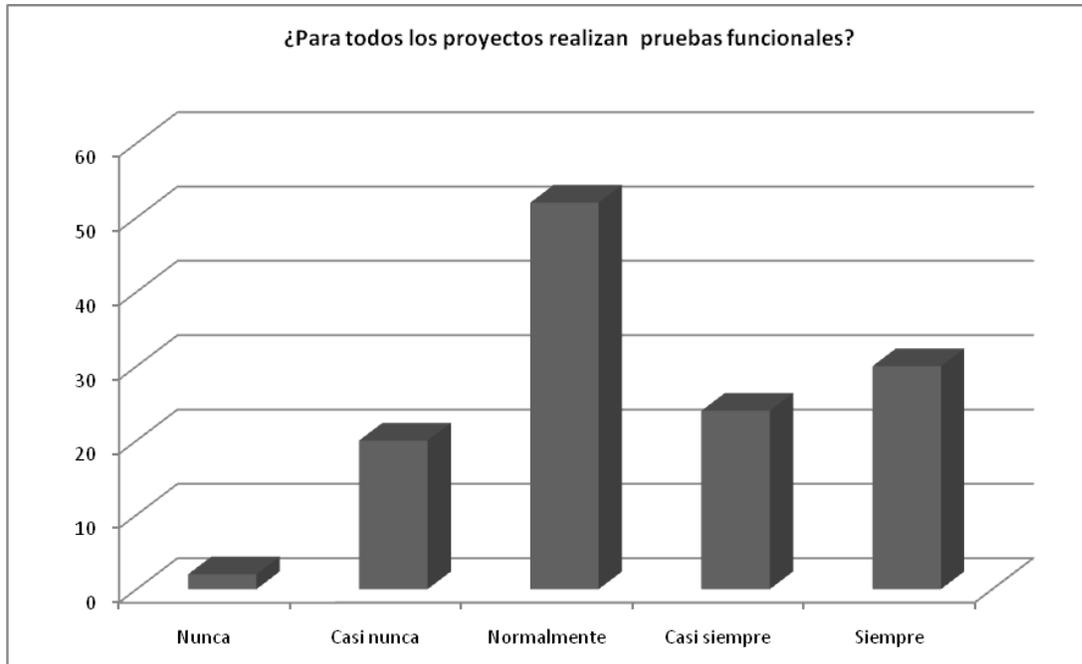
1.6.2 De los tipos de pruebas

La segunda parte de la encuesta obtiene información de la utilización de los diferentes tipos de pruebas que existen

1.6.2.1 Para todos los proyectos realizan pruebas funcionales

Basadas en las funciones (lo que el sistema hace) y características (descritas en documentos y comprendidas por los testers). Considera el comportamiento externo del sistema (pruebas de caja negra).

Figura 6. Pruebas funcionales



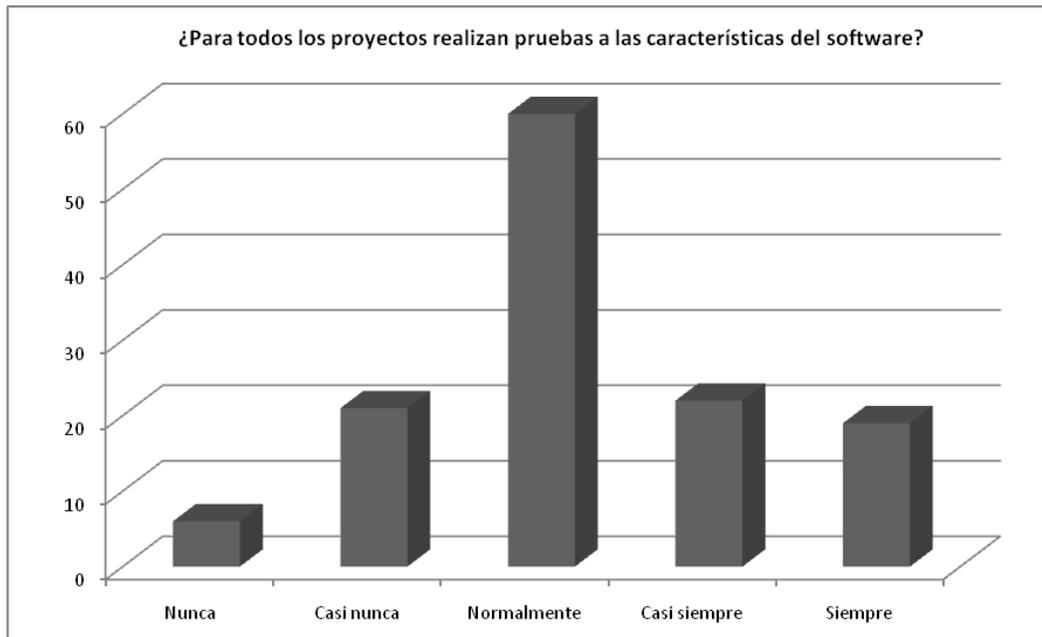
Al equipo encargado del aseguramiento de la calidad le interesa que el sistema realice las funciones que están descritas en los documentos donde se describen los requerimientos. Por esa razón, normalmente se aplican este tipo de pruebas.

El 40.63% de los encuestados respondieron de que normalmente realizan pruebas funcionales, el 18.75% casi siempre realiza pruebas funcionales y el 23.44% realiza este tipo de pruebas por lo que la tendencia es de que casi siempre se realizan las pruebas funcionales al software.

1.6.2.2 Para todos los proyectos realizan pruebas a las características del software

Son las pruebas de cómo trabaja el sistema. Estas pruebas requieren de características cuantificables de los sistemas y el software que pueden ser cuantificables en una escala variable, como por ejemplo los tiempos de respuesta para las pruebas de rendimiento.

Figura 7. Pruebas a las características del software



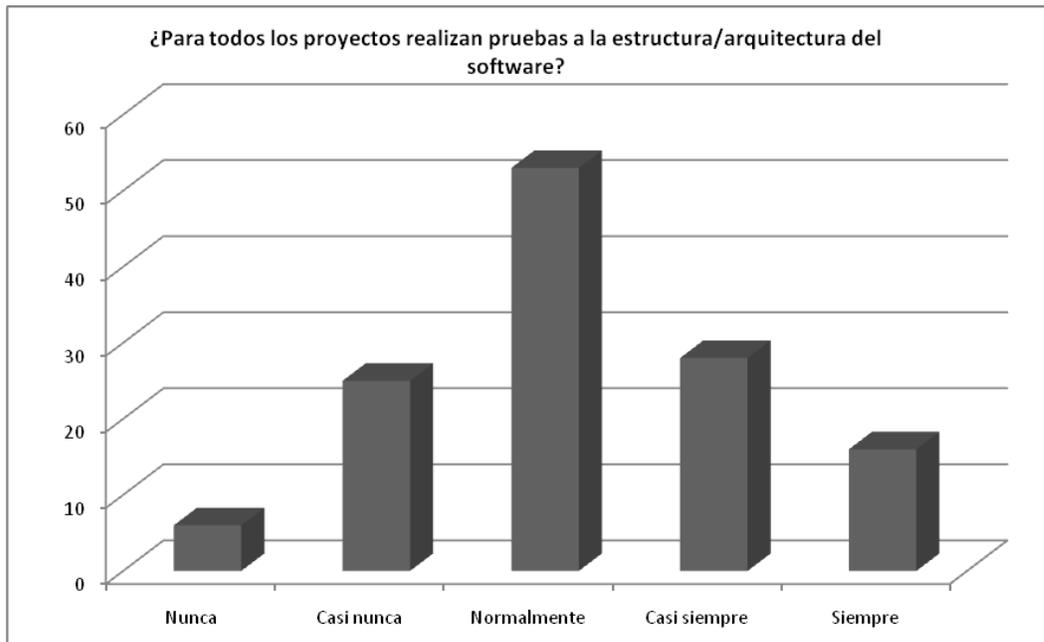
El 46.88% de las personas encuestadas normalmente realizan pruebas a las características del software, el 16.41% casi nunca realiza estas pruebas, 17.19% casi siempre realiza las pruebas.

Esta gráfica nos muestra que la tendencia es que normalmente se realizan las pruebas a las características del software.

1.6.2.3 Para los proyectos que requieren diseño del sistema, realizan pruebas a la estructura/arquitectura del software

Pruebas estructurales (pruebas de caja blanca) pueden realizarse en todos los niveles de pruebas. Especialmente en las pruebas de integración de componentes. Estas pruebas se pueden basar en la arquitectura del sistema.

Figura 8. Pruebas a la estructura del software



La tendencia de las pruebas a la estructura del software es que normalmente se realicen este tipo de pruebas.

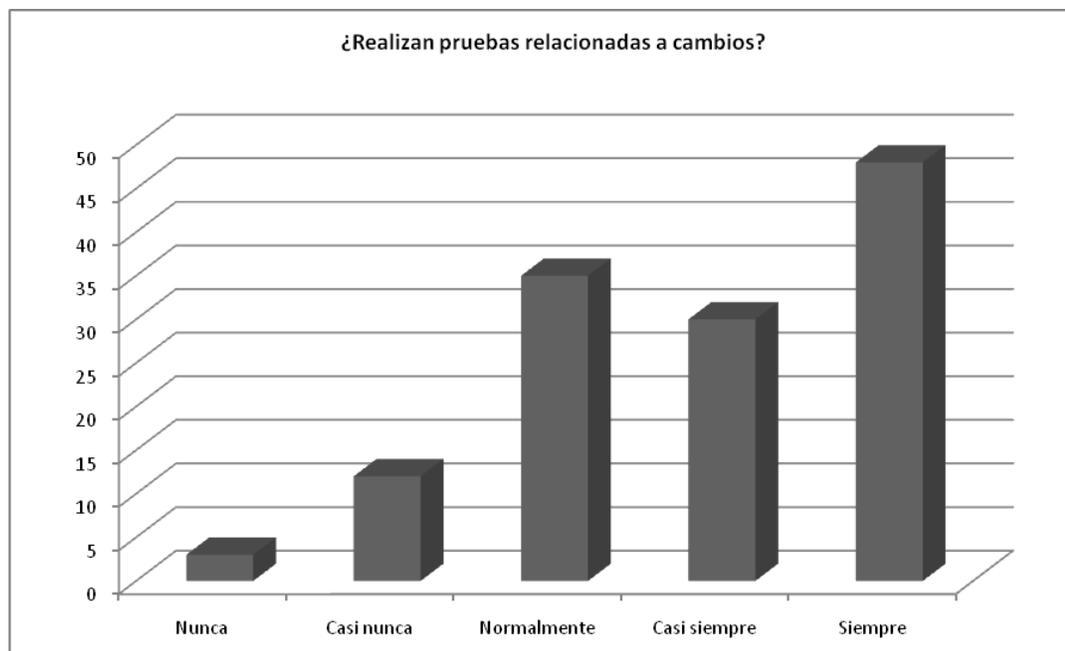
Podemos observar que el 19.53% casi nunca realiza estas pruebas, luego el 41.41% normalmente las realiza, y un 21.88% casi siempre ejecuta este tipo de pruebas. Estas pruebas se realizan normalmente pero en la gráfica podemos observar que la cantidad de personas que aplican estas pruebas se encuentran

distribuidas en que casi nunca y casi siempre realizan este tipo de pruebas. Por lo que podemos decir que se aplican normalmente pero con menos frecuencia que las pruebas a las características del software.

1.6.2.4 Realizan pruebas relacionadas a cambios

Cuando se detecta un defecto y es arreglado, el software debe ser probado nuevamente para confirmar que el defecto haya sido removido.

Figura 9. Pruebas de cambios al software

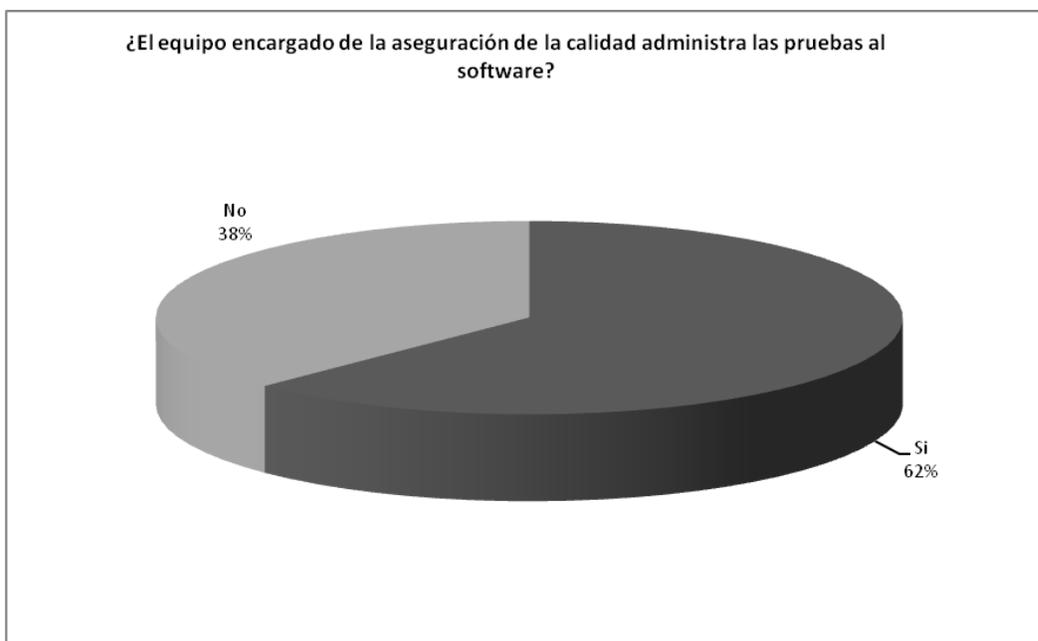


Este tipo de pruebas tiene una gráfica muy diferente a los demás tipos de pruebas, como podemos observar la mayoría de encuestados siempre realizan este tipo de pruebas. El 37.50% de las personas encuestadas siempre realizan pruebas relacionadas a cambios, este alto porcentaje refleja que es de suma importancia asegurarse de que el defecto del software ha sido corregido

satisfactoriamente, el 23.44% casi siempre realiza estas pruebas pero es mayor el porcentaje que normalmente las realiza con 27.34%, aunque es mínimo el porcentaje de los encuestados que nunca realizan pruebas relacionadas a cambios con el 2.34% debería de ser 0% para verificar que no se ha creado otro defecto cuando se corrige uno.

1.6.2.5 ¿El equipo encargado de la aseguración de la calidad administra las pruebas realizadas?

Figura 10. Administración de las pruebas de software



Este resultado muestra que el 62% de las personas encuestadas saben que el equipo encargado del aseguramiento de la calidad del software lleva el control y la administración de cada una de las pruebas efectuadas a los diferentes sistemas que se puedan tener, llevando registros de las actividades realizadas y de los resultados obtenidos en cada una de las pruebas.

Estos resultados reflejan que las pruebas son ejecutadas correctamente por la mayoría de los equipos que aseguran la calidad del software, aunque esta cantidad puede ser aceptable esta debería de ser el 100%.

1.6.3 De las herramientas

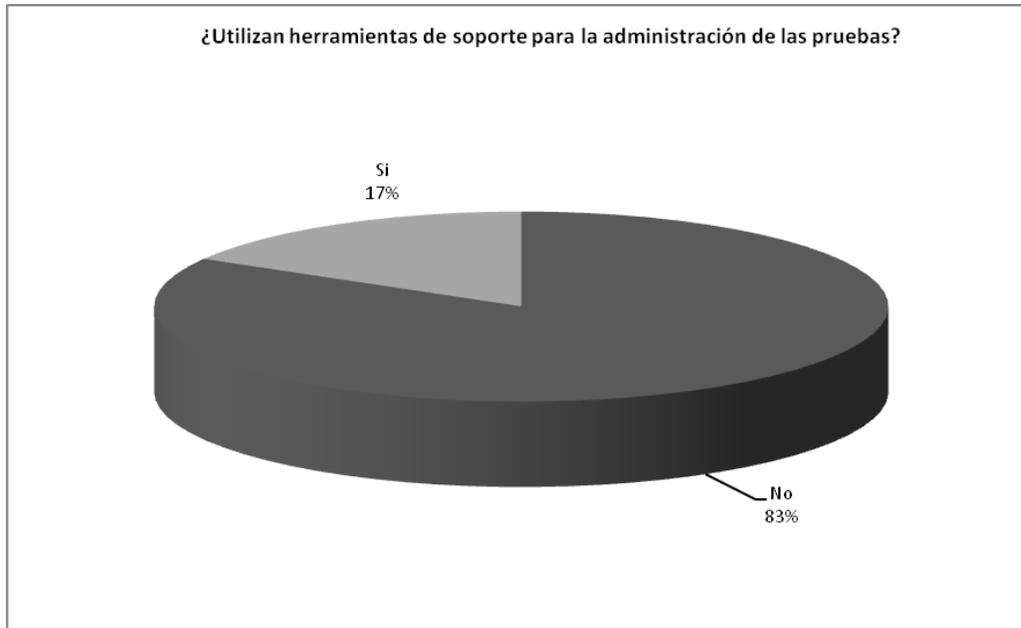
La última parte obtiene información acerca de las herramientas que utilizan para realizar las pruebas.

La utilización de las herramientas nos reflejaran el nivel del aseguramiento de la calidad del software se da por parte de las personas encuestadas.

1.6.3.1 Utiliza herramientas de soporte para la administración de las pruebas y el examen

Administración de los requerimientos, Administración de los incidentes, Administración de la configuración

Figura 11. Herramientas para la administración de pruebas de software



De las personas encuestadas, el 17% utiliza alguna herramienta de soporte para la administración de las pruebas.

Algunas de las herramientas utilizadas:

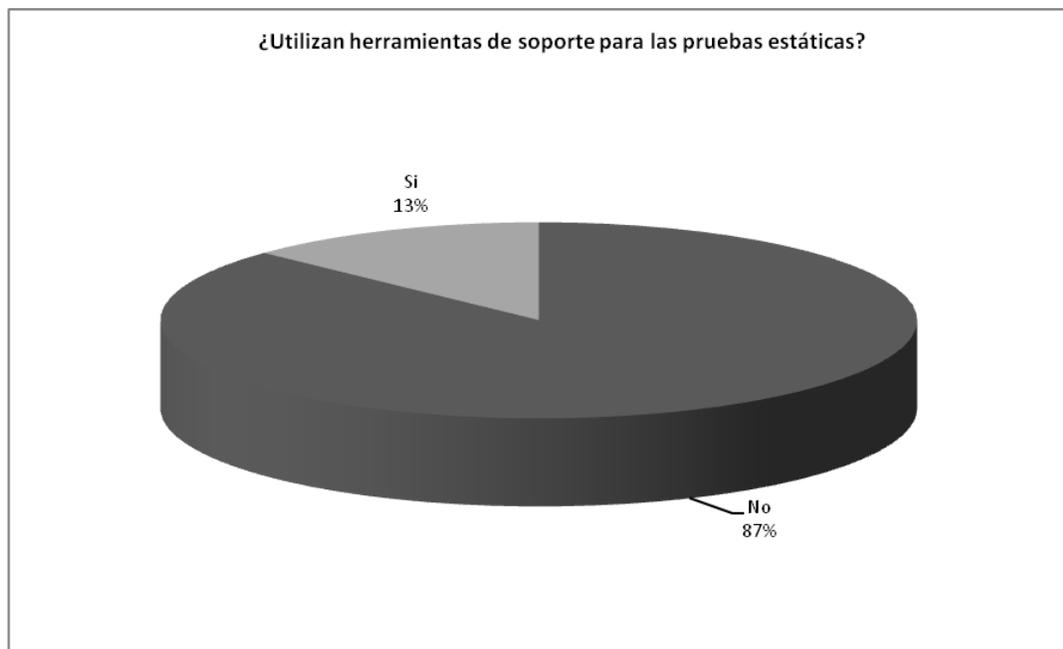
- Java
- Visual Source Safe
- dotNET ModelKit Suite
- CVS
- Sistema de control interno, desarrollado internamente
- Rational
- SCM EM ELP
- Bugzilla
- Sigma7, Excel
- VISUAL STUDIO TEAM SUITE

- JUNIT
- TechExcel (Service Wise y devtrack).
- ClearQuet

1.6.3.2 Utiliza herramientas de soporte para las pruebas estáticas

Análisis estático para apoyar los estándares de desarrollo, el análisis de las estructuras y dependencias y poder ayudar al entendimiento del código. Herramientas de modelado, las cuales validan los modelos del software.

Figura 12. Herramientas para pruebas estáticas



El 13% de las personas encuestadas utilizan alguna herramienta de soporte para las pruebas estáticas.

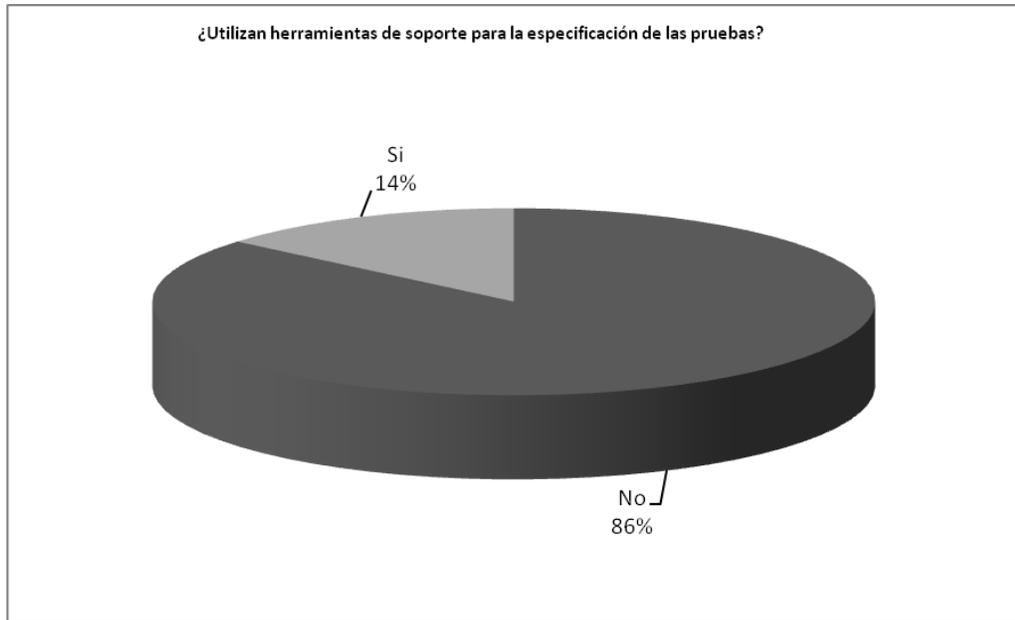
Algunas de las herramientas utilizadas:

- Visio
- dotNET ModelKit Suite
- Enterprise Architect
- Data Model, CMM, Designer
- VISIO ENTERPRISE ARCHITECT
- RATIONAL ROSE
- Diagramas en UML

1.6.3.3 Utiliza herramientas de soporte para la especificación de las pruebas

Herramientas para el diseño de pruebas, herramientas para la preparación de los datos de prueba

Figura 13. Herramientas para la especificación de pruebas



El 14% de los encuestados utiliza alguna herramienta de soporte para la especificación de las pruebas.

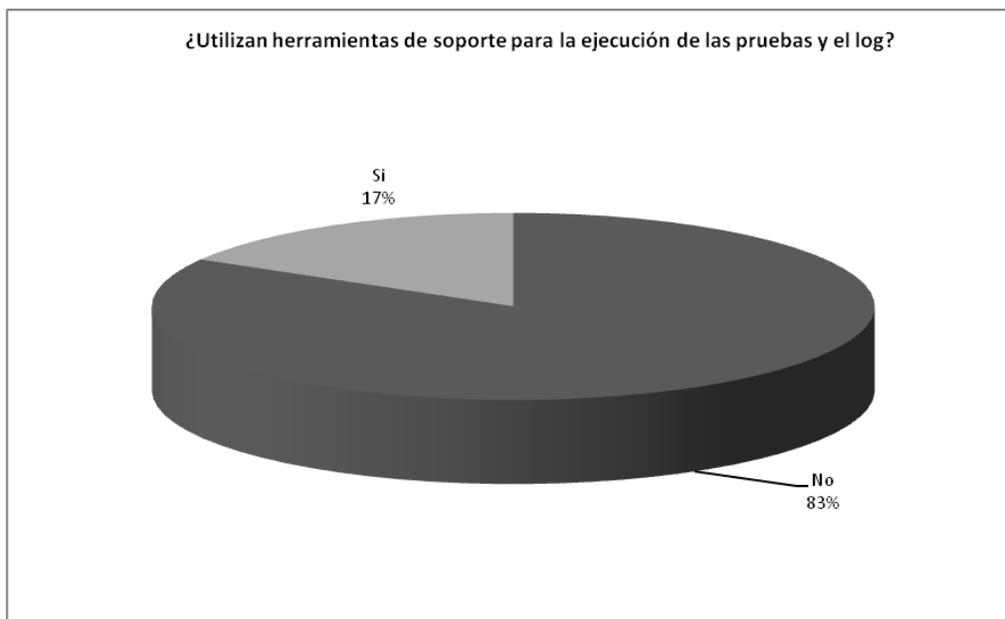
Algunas de las herramientas utilizadas:

- dotNET ModelKit Suite
- Clear Quest
- Rational
- SCM
- faces
- Rational Test Manager
- VISUAL STUDIO TEAM SUITE, JUNIT
- Administración de pruebas - RUP
- Test manager

1.6.3.4 Utiliza herramientas de soporte para la ejecución de las pruebas y el log

Herramientas para la ejecución de pruebas, herramientas que simulan el ambiente, en el cual el objeto a probar se ejecutará, comparadores de pruebas y herramientas de seguridad

Figura 14. Herramientas para la ejecución de pruebas



El 17% de las personas encuestadas utilizan herramientas de soporte para la ejecución de las pruebas y poder llevar el control de los resultados obtenidos almacenándolos en una bitácora.

Algunas de las herramientas utilizadas:

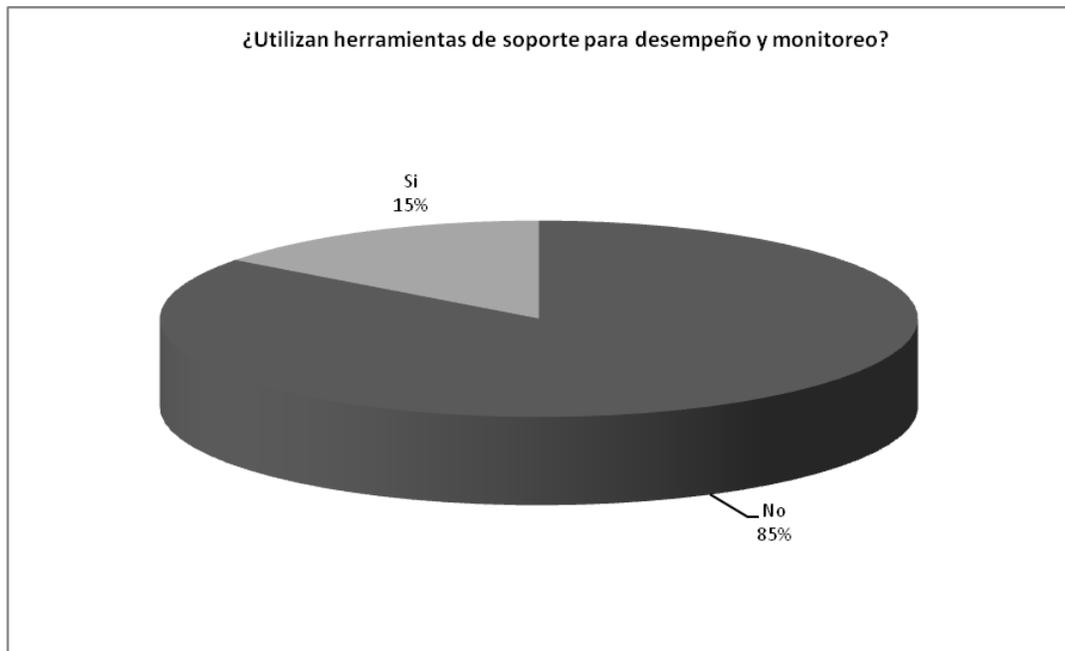
- VMware virtual machine
- dotNET ModelKit Suite
- JUnit

- log4java
- VISUAL STUDIO TEAM SUITE
- Nunit- Microsoft Visual Studio
- Notepad

1.6.3.5 Utiliza herramientas de soporte para desempeño y monitoreo

Herramientas de análisis dinámico, herramientas para pruebas de estrés y herramientas de monitoreo.

Figura 15. Herramientas para monitoreo



El 15% de los encuestados utilizan alguna herramienta de soporte para el desempeño y monitoreo de los sistemas de software que se desarrollan.

Algunas de las herramientas utilizadas:

- web application stress
- dotNET ModelKit Suite
- EM
- Netbeans Profiler
- VISUAL STUDIO TEAM SUITE
- Jmeter
- Vantage

2. MERCADO

Con el estudio de la oferta y la demanda se puede saber si existe una demanda insatisfecha que justifique la implantación de un Centro de Pruebas de Software. Históricamente no se tienen datos explícitos de la oferta, ya que en Guatemala actualmente no existen proveedores de los servicios de pruebas de software.

La demanda está determinada por factores como el precio del bien o servicio, los precios de los sustitutos y de sus complementarios, la renta personal y las preferencias individuales del consumidor.

2.1 Estudio de mercado

Se ha realizado un estudio de mercado, el cual evalúa las necesidades reales de los clientes actuales y potenciales. Se estima que el mercado del Centro de Pruebas será nacional con proyecciones centroamericanas.

2.1.1 Tamaño del mercado

Empresas de desarrollo de software con miras a exportación de sus productos.

La metodología de la encuesta fue pasar el cuestionario por correo electrónico hacia el contacto publicado en la página principal de SOFEX, enviando de regreso el cuestionario contestado

Segmento del mercado

En el corto plazo el segmento del mercado elegido por el centro de pruebas, es el de las empresas de desarrollo de soluciones de software de Guatemala que estén asociadas a SOFEX (Comisión de software de Guatemala).

Se eligió este segmento de mercado debido a que los objetivos de SOFEX se encuentran alineados con los objetivos y metas del centro de pruebas, principalmente el objetivo “Apoyar procesos de mejora continua aumentando la calidad de los productos de Software guatemalteco” [5].

2.1.2 Consumo aparente

Resulta de la sumatoria de la producción Nacional (pruebas realizadas por empresas nacionales) más las importaciones (pruebas realizadas por empresas extranjeras a empresas nacionales) menos las exportaciones (pruebas realizadas por empresas nacionales a empresas extranjeras), lo cual nos indica la disponibilidad de producto para satisfacer la demanda Nacional.

Las pruebas de software son realizadas en todo proyecto de desarrollo de software. Cada empresa que desarrolla software normalmente realiza tres tipos de pruebas en los proyectos que realiza siendo estas las pruebas funcionales,

pruebas a las características del software y pruebas a la estructura y arquitectura del software.

Del estudio de mercado en promedio cada cliente realiza 8 proyectos nuevos de desarrollo de software al año.

2.1.3 Demanda potencial histórica

Históricamente no se tienen datos de la demanda de los servicios como tal, pero como vimos en el capítulo del “Nivel de pruebas en Guatemala”, los servicios de pruebas son normalmente utilizados en el desarrollo de todo proyecto.

Las empresas que pertenecen a SOFEX son los mejores candidatos a usuarios de los servicios del centro de pruebas.

2.1.4 Demanda insatisfecha

Es la parte de la población que no encontró la satisfacción de sus necesidades ante la escasa oferta. Debido a que no hay oferta en los servicios, toda la demanda queda insatisfecha.

2.1.5 Objetivos del estudio de mercado

- Conocer al cliente
- Definir el producto
- Conocer la frecuencia y consumo del producto

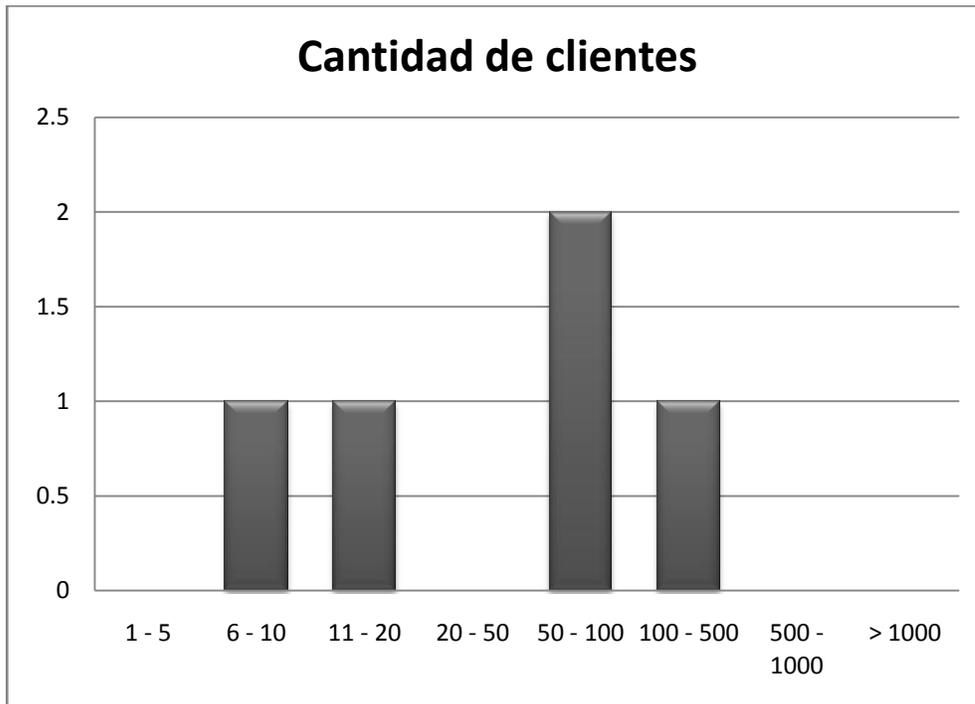
- Conocer la aceptación del producto en el mercado
- Que características del producto prefieren los clientes
- Cantidad de producto a consumir y cuanto están dispuestos a pagar
- Que problemas se tienen con el control de calidad del software
- Demostrar la factibilidad de abrir un centro de pruebas para software en el país

2.1.6 Resultados obtenidos

Los resultados obtenidos de la encuesta (ver anexo Encuesta II) son los siguientes:

1. Cantidad de clientes:

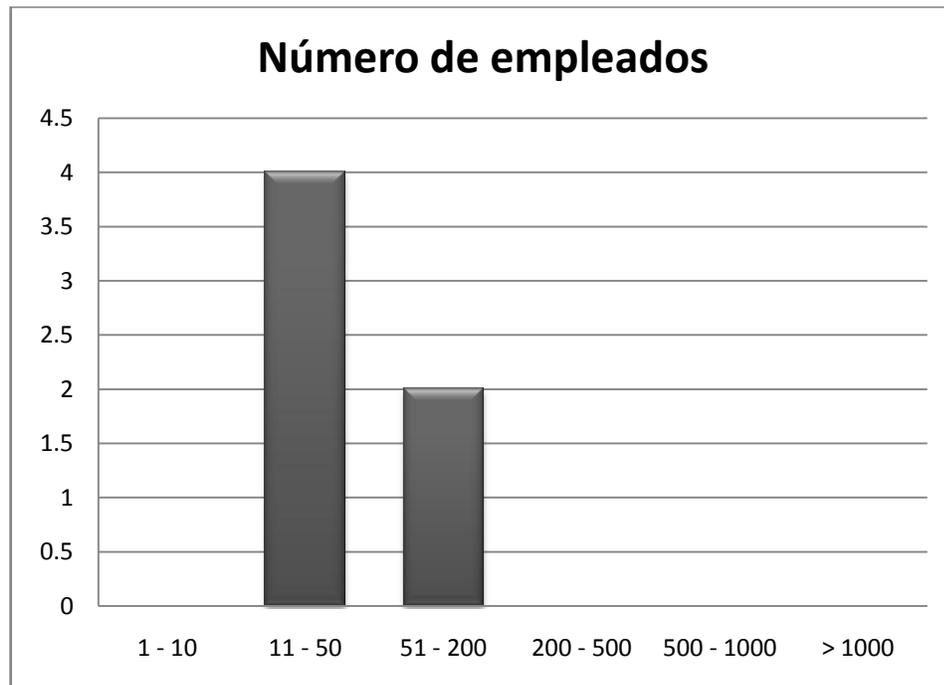
Figura 16. Cantidad de clientes



La cantidad de clientes nos da una idea del tamaño de la empresa y el impacto que tiene esta en el mercado, las empresas encuestadas están en un rango entre empresas medianas y pequeñas, pues la cantidad de clientes está por debajo de los 500 clientes y en su mayoría tienen más de 50 y 100 clientes.

2. Número de empleados:

Figura 17. Número de empleados



El tamaño de una empresa puede ser medido también por la cantidad de empleados contratados, en este caso los empleados contratados oscilan en su mayoría entre los 11 y 50 empleados. Esto muestra que las empresas encuestadas son más en su mayoría medianas empresas.

3. ¿Contrata actualmente servicios externos de control de calidad y pruebas de software?

Figura 18. ¿Contrata servicios de control de calidad y pruebas de software?



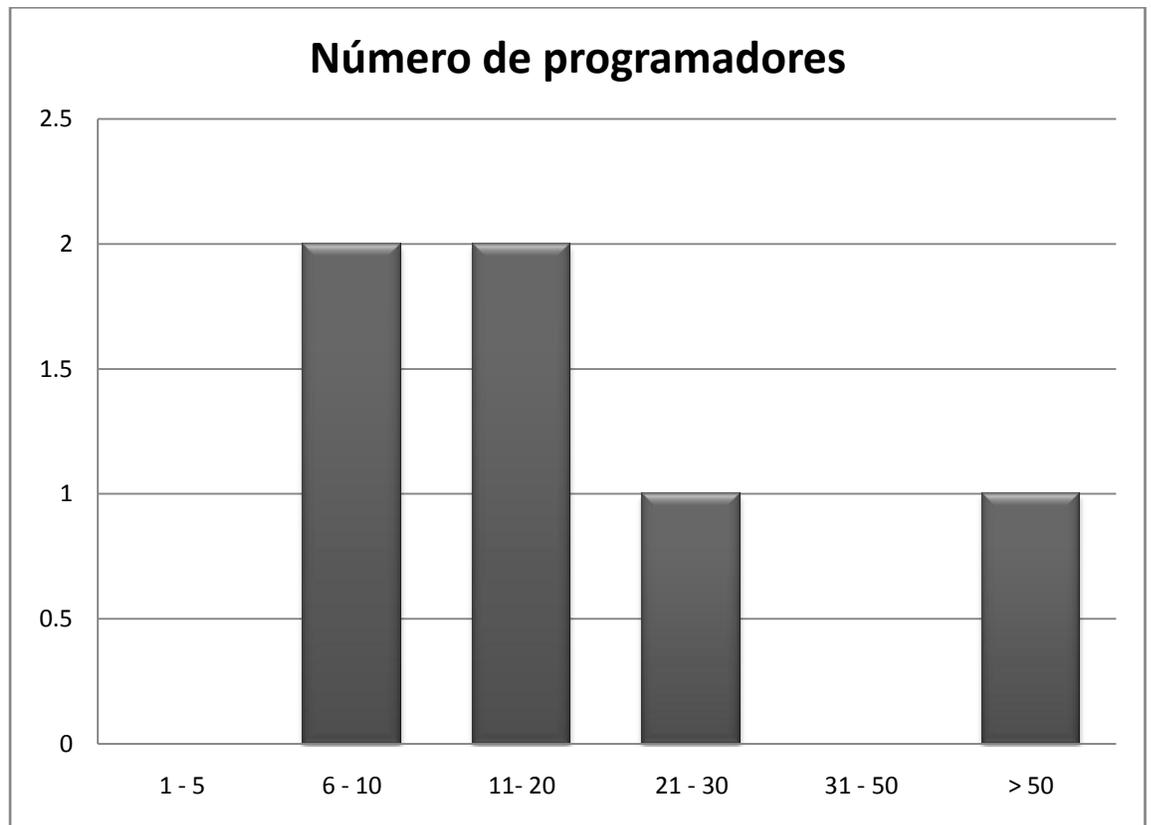
La gráfica muestra como en Guatemala actualmente no se contratan servicios de calidad de un ente externo, ya que el 100% de los encuestados respondió que no contrata ningún ente externo a la empresa para comprobar la calidad de sus productos.

4. ¿De qué proveedor contrata servicios de control de calidad y pruebas de software?

No existe proveedor de servicios de control de calidad y pruebas de software para las empresas.

5. Número de programadores

Figura 19. Número de programadores



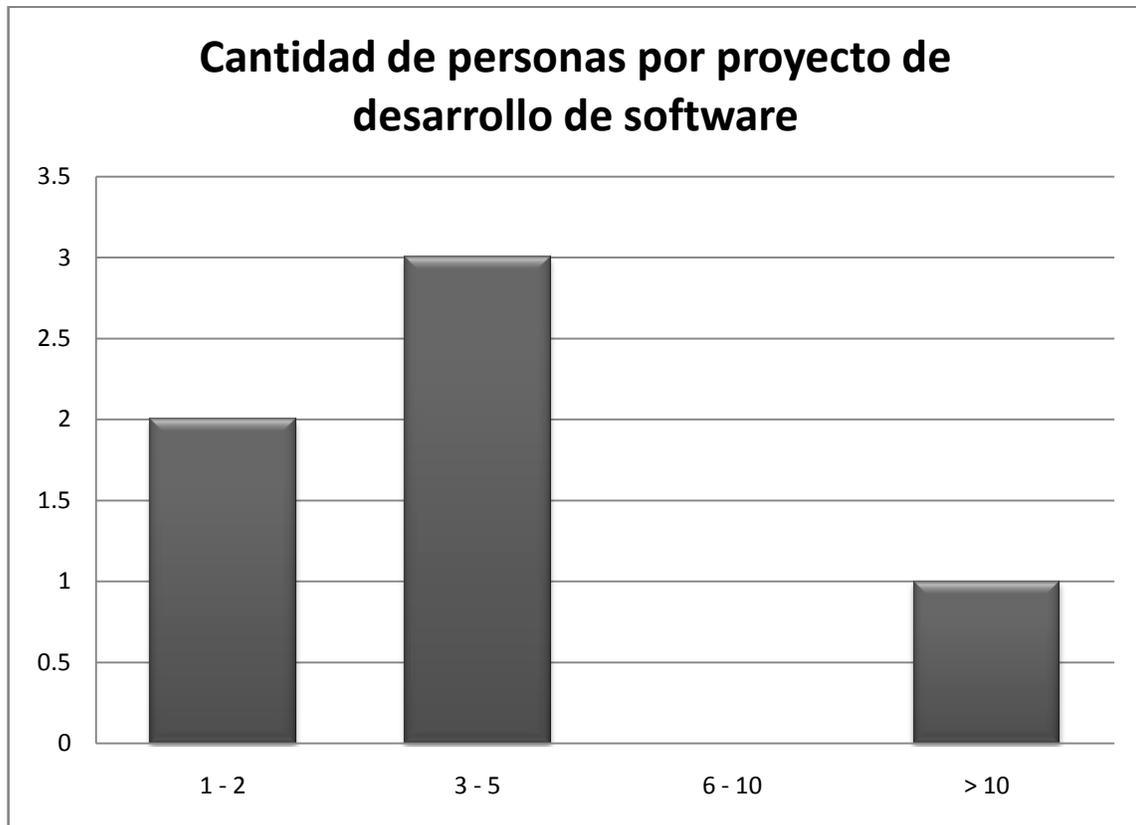
Un mayor porcentaje de las empresas encuestadas tienen entre 6 y 20 programadores. En menor porcentaje hay empresas que tienen entre 21 y 30 programadores así como más de 50.

6. Cantidad de proyectos nuevos de desarrollo de software al año:

En promedio tienen 9 proyectos nuevos al año

7. Cantidad de personas por proyecto de desarrollo de software

Figura 20. Personas por proyecto



La cantidad de personas designadas a un proyecto por año, indica la magnitud de los proyectos. Las empresas encuestadas muestran que la cantidad de personas designadas por proyecto es pequeña, por lo que se puede decir que el tamaño de los proyectos es pequeño o mediano.

8. Cantidad de productos de software para venta que poseen:

En promedio las empresas de la muestra poseen 12 productos de software para la venta

9. Realizan desarrollo de software a la medida:

Figura 21. Desarrollo de software a la medida



El 83% de las empresas realizan desarrollo de software a la medida dejando un 17% de las empresas que se dedican a realizar productos de software específicos.

10. Edad del producto líder

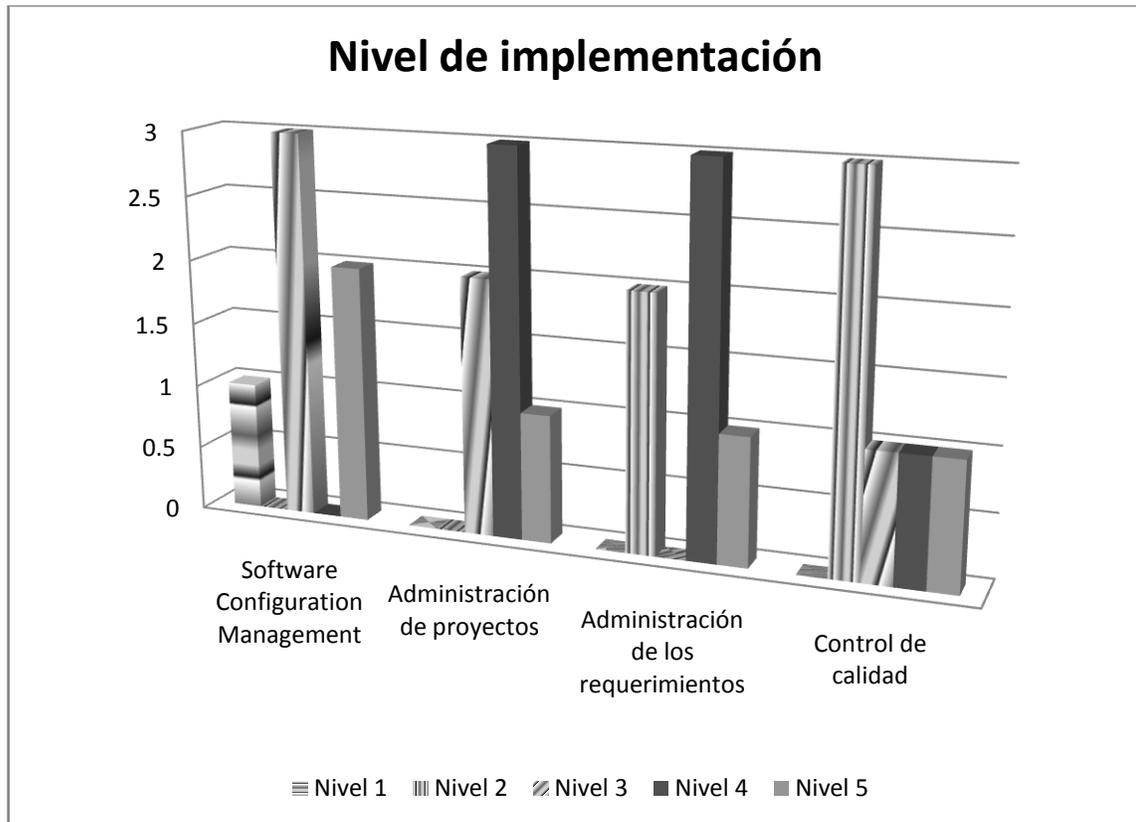
En promedio 10 años tiene el producto líder de las empresas

11. Versión del producto líder

La versión del producto líder varía de la 4 a la 6, en promedio los productos líderes estarán en la versión 5

12. Nivel de implementación en su empresa de las siguientes prácticas.

Figura 22. Nivel de implementación



Para el *Software Configuration Management*, la mayoría de las empresas tienen un nivel 3 de implementación, seguido de un nivel 5, y en menor cantidad está el nivel 1, dejando al nivel 2 y 4 de implementación sin participación.

Para la administración de proyectos el nivel 4 es el más implementado, seguido del nivel 3 y de último el nivel 5. Estas empresas tienen una buena administración de los proyectos ya no hay empresas por debajo del nivel 3.

En la administración de los requerimientos el nivel 4 predomina, seguido por el nivel 2. En un menor porcentaje de participación está el nivel 5.

En cuanto al control de calidad de los productos de software el mayor nivel de aplicación es el nivel 2. Para el nivel 1 no existen registros de aplicación, sin embargo para el nivel 3, 4 y 5 existe el mismo nivel de aplicación.

13. Interés en adquirir los servicios de control de calidad de software de una empresa externa para productos nuevos de la empresa

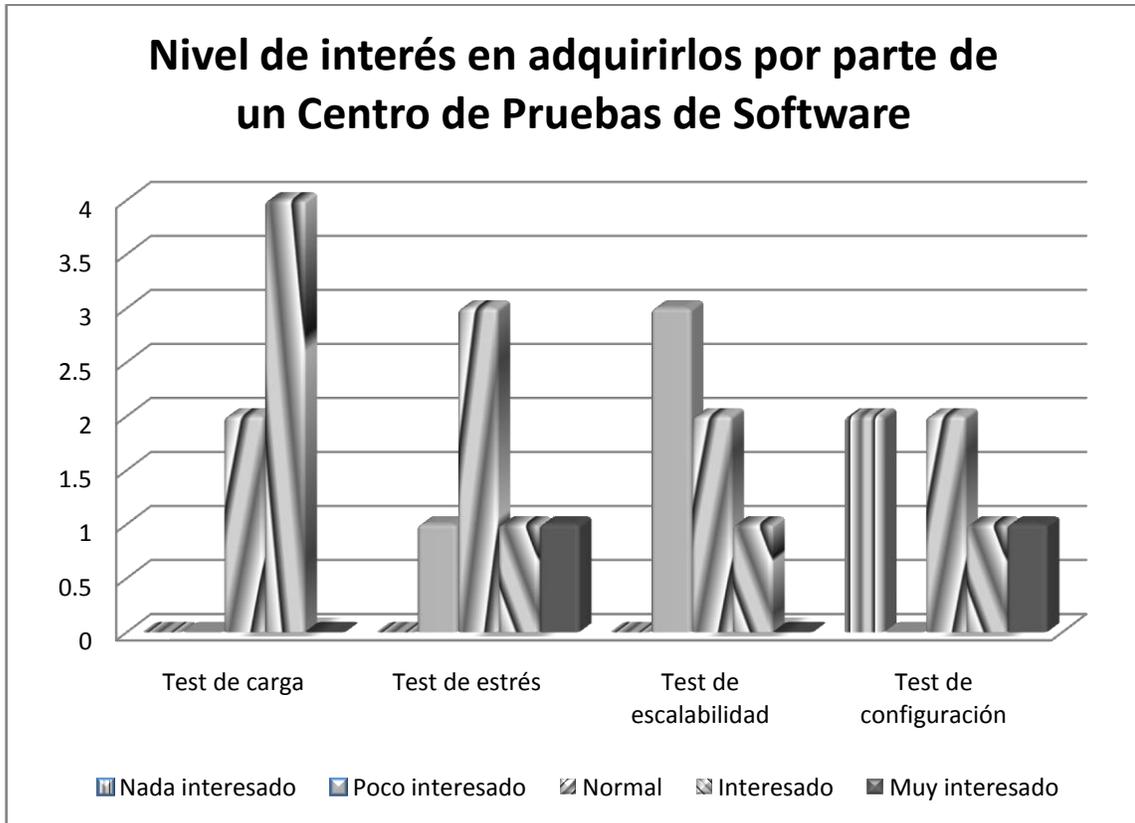
En promedio las empresas tienen un nivel de interés en adquirir los servicios de 5 puntos de 10.

14. Porcentaje de errores reportados al departamento de desarrollo de software en las solicitudes de soporte

En promedio 23% de las solicitudes de soporte corresponden a errores de los productos.

15. De la siguiente lista de ensayos de plataforma, asigne el nivel de interés en adquirirlos por parte de un Centro de Pruebas de Software.

Figura 23. Interés de ensayos de plataforma



El 66% de los encuestados están interesados en adquirir los servicios para un *test* de carga y el 33% se mostró indiferente.

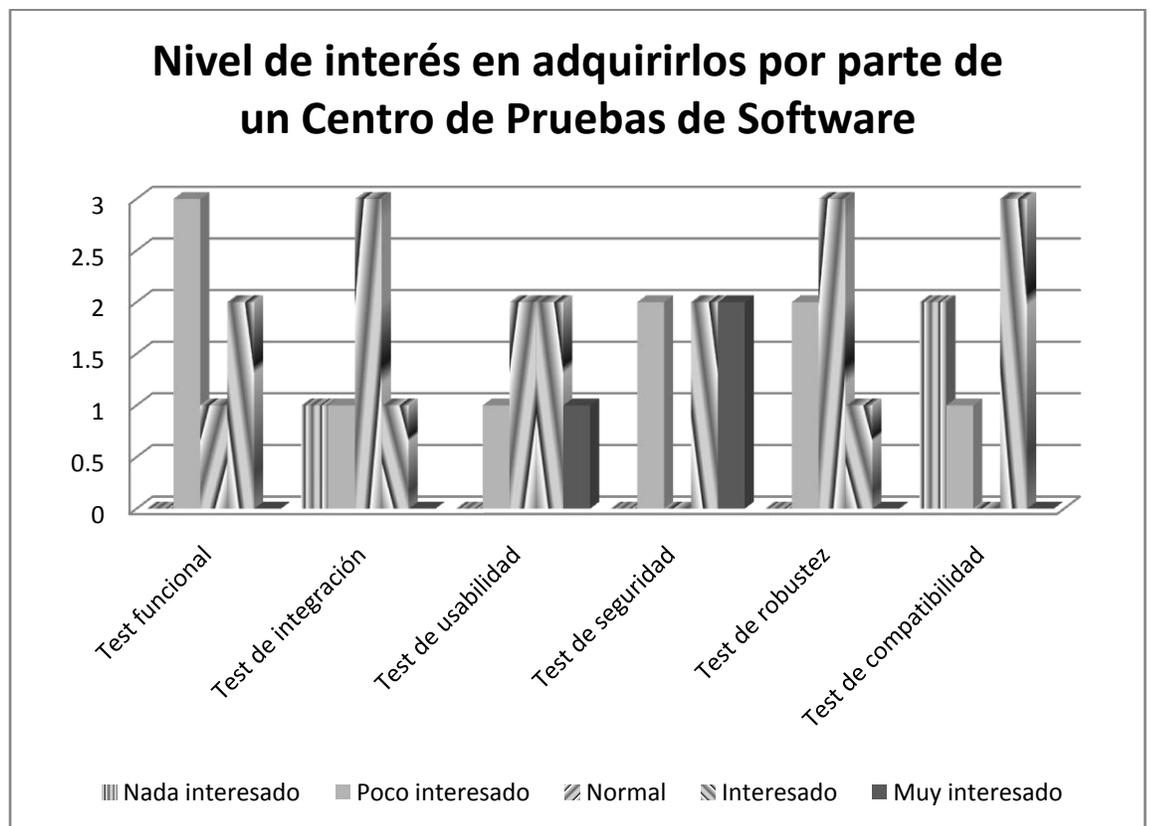
En cuanto a los servicios del *test* de estrés un 50% de los encuestados se mostró indiferente, un 16% poco interesado, 16% interesado y otro 16% muy interesado.

El 50% de los encuestados mostró poco interés en los servicios de *test* de escalabilidad, un 33% se mostró normal y un 16% interesado.

Para el *test* de configuración, el 33% se mostró nada interesado, otro 33% se mostró indiferente mientras que el 16% se mostró nada interesado y otro 16% muy interesado.

16. De la siguiente lista de servicios de pruebas, asigne el nivel de interés en adquirirlos por parte de un Centro de Pruebas de Software.

Figura 24. Interés de servicios de pruebas



La mayoría de los encuestados están poco interesados en adquirir servicios de *test* funcional, el 50% está poco interesado, el 16% normal y otro 33% interesado.

Para los servicios de *test* de integración, 50% está normal, 16% está interesado, 16% poco interesado y otro 16% nada interesado.

El 33% está interesado en las pruebas de usabilidad, otro 33% normal, un 16% poco interesado y otro 16% muy interesado.

Para el *test* de seguridad la distribución entre los encuestados fue muy pareja quedando un 33% poco interesado, 33% interesado y otro 33% muy interesado.

Para los servicios de *test* de robustez, el 50% tiene un nivel de interés normal en adquirirlos, un 33% está poco interesado, y un 16% interesado.

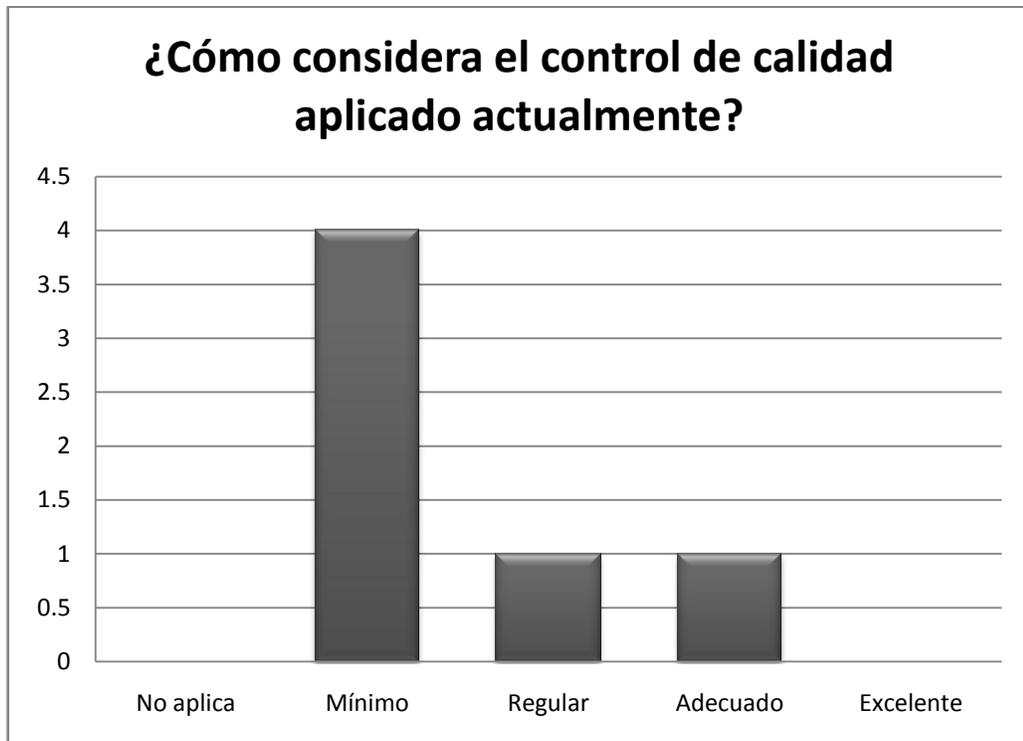
El 50% de las empresas están interesadas en adquirir los servicios de *test* de compatibilidad, un 33% nada interesado, y un 16% poco interesado.

17. Porcentaje del presupuesto por proyecto asignado a Control de Calidad

En promedio se asigna el 18% del presupuesto del proyecto al control de calidad.

18. ¿Cómo considera el control de calidad aplicado actualmente?

Figura 25. Control de calidad actual



El 66% de los encuestados consideran que el control de calidad aplicado actualmente es mínimo, un 16% considera que es adecuado y otro 16% considera que es regular.

19. Nombre algunas de las empresas que provean servicios de pruebas de software en su región

No se tiene conocimiento de que exista alguna empresa que provea los servicios de pruebas de software en la región.

2.1.7 Conclusiones del estudio realizado

- La mayoría de empresas de software guatemaltecas se les puede catalogar como pequeñas empresas, tal como lo muestran los resultados de las preguntas 1 y 2. De la misma manera las preguntas 5 y 7 nos dan una idea del tamaño de los proyectos que se manejan a nivel regional, estos proyectos parecen ser de tamaño mediano, considerando que entre menos gente asignada tengan, menor será su tamaño. La cantidad de proyectos que se manejan anualmente por empresa de software nos indica que en el país se encuentra en constante movimiento las ventas en la industria de software.
- En Guatemala no se utilizan ni se conocen empresas dedicadas al control de calidad de software, por lo que un centro de pruebas, podría colaborar en gran medida a dar soporte formal a las empresas acerca de la calidad de sus productos.
- La mayoría de empresas guatemaltecas de software, realizan desarrollo a la medida. El desarrollo a la medida, reincide en mayor cantidad en errores, tanto del lado de los programadores como de los analistas, diseñadores y administradores, puesto que este requiere de gran trabajo y adaptación al negocio. Este tipo de desarrollo requiere de mayor cantidad de pruebas que los que no son a la medida del cliente, puesto que tienen que desarrollarse nuevos requerimientos en distintos clientes y por lo tanto nuevas pruebas, implementaciones, etc.
- El nivel de implementación de algunas prácticas de software no están en el mayor nivel, el centro de pruebas puede ayudar a elevar el nivel de implementación de esas prácticas.

- De los ensayos de plataforma, el *test* de carga es el servicio que tiene más oportunidad por parte del centro de pruebas
- De los servicios de pruebas, el *test* de usabilidad, el *test* de seguridad y el *test* de compatibilidad son los que tienen mayor oportunidad por parte del centro de pruebas de software.
- Tanto el presupuesto como el control de calidad aplicado actualmente es bajo. La estrategia de entrada del centro de pruebas de software sería de proveer servicios de control de calidad especializados a un precio accesible.

2.2 Promoción del producto o servicio

El proceso de comercialización de los servicios del centro de pruebas se realizará mediante la mezcla de los elementos principales de la mercadotecnia: promoción, plaza o distribución, producto y precio. ^[6]

2.3 Fijación y políticas de precio

El precio de un servicio de pruebas de software es calculado luego de un análisis y estimación del tiempo de duración de la prueba.

El precio por hora de cada integrante del proyecto está dado en la siguiente tabla:

LIDER DE PROYECTO	US \$ 24
CONSULTOR	US \$ 20
DISEÑADOR DE CASOS DE PRUEBA	US \$ 14
TESTER	US \$ 10

2.4 Riesgos y oportunidades del mercado

Riesgos	Oportunidades
Falta de presupuesto asignado a la etapa de pruebas en los proyectos.	No existe otra empresa que se dedique a cubrir esta necesidad.
Que las empresas no vean la necesidad de contratar servicios de pruebas como algo adicional.	Facilidad de cobertura y promoción por medio de SOFEX.
Que las pruebas formales se realicen por personas internas a la empresa o por los mismos clientes.	Necesidad de tener un soporte formal que proporcione control de calidad al software en la región.
Falta de madurez del proceso de arquitectura de software.	

3. PRODUCCIÓN

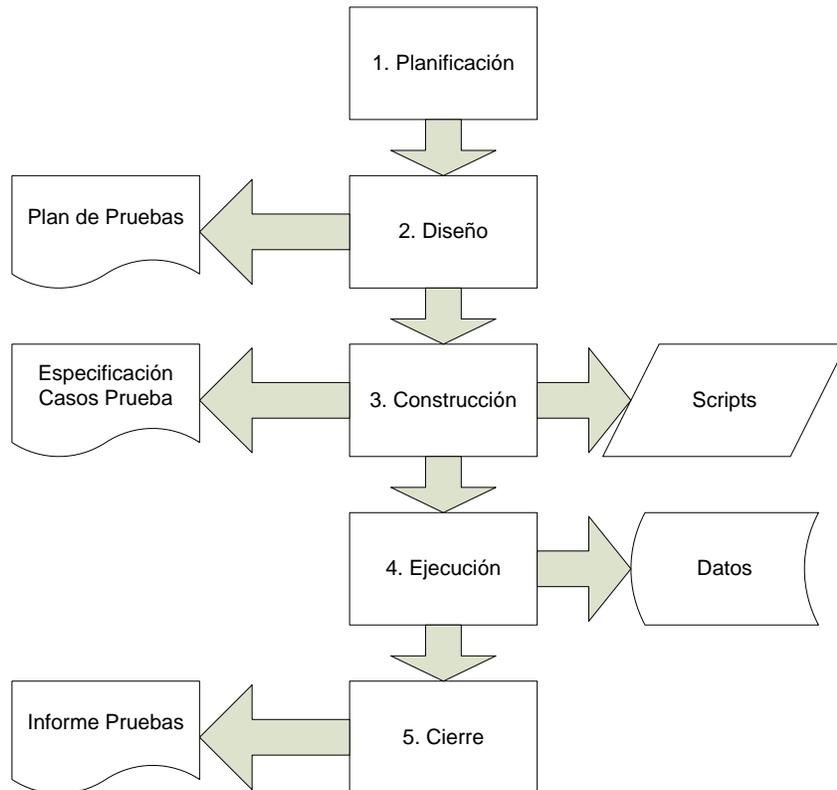
El principal objetivo de las pruebas es mejorar la calidad encontrando las debilidades y fortalezas de los sistemas de información, los beneficios que se obtienen no son únicamente mejorar la calidad de los productos y aplicaciones sino que también se reducen los esfuerzos en mantenimientos, además de mejorar la satisfacción de los usuarios y clientes.

3.1 Descripción del proceso productivo o prestación del servicio

Existen algunos otros procesos relacionados a las pruebas de software que ayudan al desarrollo de la calidad del producto como lo son la ingeniería de requisitos, la calidad de código, la gestión de proyectos y la gestión de la configuración.

3.1.1 Las pruebas como proceso

Figura 26. Diagrama de flujo de las pruebas como proceso



El proceso de pruebas se puede modelar en cinco etapas. Este proceso aplica cuando tenemos un producto de software terminado y se le va a aplicar algunas pruebas.

Las etapas del proceso son secuenciales y a lo largo del mismo se construyen algunos artefactos que ayudan a concluir con éxito el proceso.

A. Planificación

La primera etapa del proceso consiste en realizar una planificación de las pruebas. Las entradas principales de la planificación de las pruebas son: el plan del proyecto de software, tomando en cuenta los hitos importantes; los esfuerzos de desarrollo, considerando el tamaño del proyecto y la tecnología utilizada; y las restricciones del diseño y la arquitectura del software. Como resultado de esta etapa se obtiene un plan de pruebas el cual contiene las actividades principales de la prueba, los recursos a utilizar y las fechas límite de cada actividad, tipos de prueba a aplicar y las herramientas de pruebas.

B. Diseño

La etapa siguiente es el diseño de la pruebas, de acuerdo con el tipo de prueba solicitado por el cliente y las características del producto de software se diseña una prueba adecuada utilizando las herramientas y técnicas apropiadas como las de causa y efecto, ortogonalidad y valores límite. Las entradas principales del diseño de las pruebas son: el plan de pruebas especificando las prioridades de cada actividad; los requisitos del sistema y las interfaces, identificando el riesgo y la criticidad; una entrada complementaria para el diseño de las pruebas es la fiabilidad requerida del sistema. Con el buen diseño de una prueba se obtienen las especificaciones de los casos de prueba en los cuales se describen las acciones que se realizarán sobre el producto de software y el resultado esperado del mismo. Se verifica la documentación del sistema poniendo especial atención en la consistencia, la ambigüedad y la completitud de la misma.

C. Construcción

Las fases de construcción y ejecución de las pruebas van de la mano, las entradas principales de estas fases son: las especificaciones de las pruebas con su prioridad y riesgo; los requisitos funcionales, las interfaces y su rentabilidad; entre otras entradas importantes se encuentra la dificultad de construcción del diseño y el código del software. Las herramientas que se pueden utilizar en esta fase de construcción van desde las utilidades que pueda dar el sistema operativo, hasta cualquier aplicación open source o propietaria.

Las técnicas apropiadas nos pueden dar los scripts de ejecución que contienen los procedimientos automáticos, procesos de regresión y los escenarios de prueba, los cuales simularán la interacción del usuario final con el producto de software.

D. Ejecución

En la fase de ejecución de la prueba se obtienen todos los datos de respuesta los cuales se almacenan para su posterior análisis. Estos juegos de datos son reusables, completos y exactos.

E. Cierre

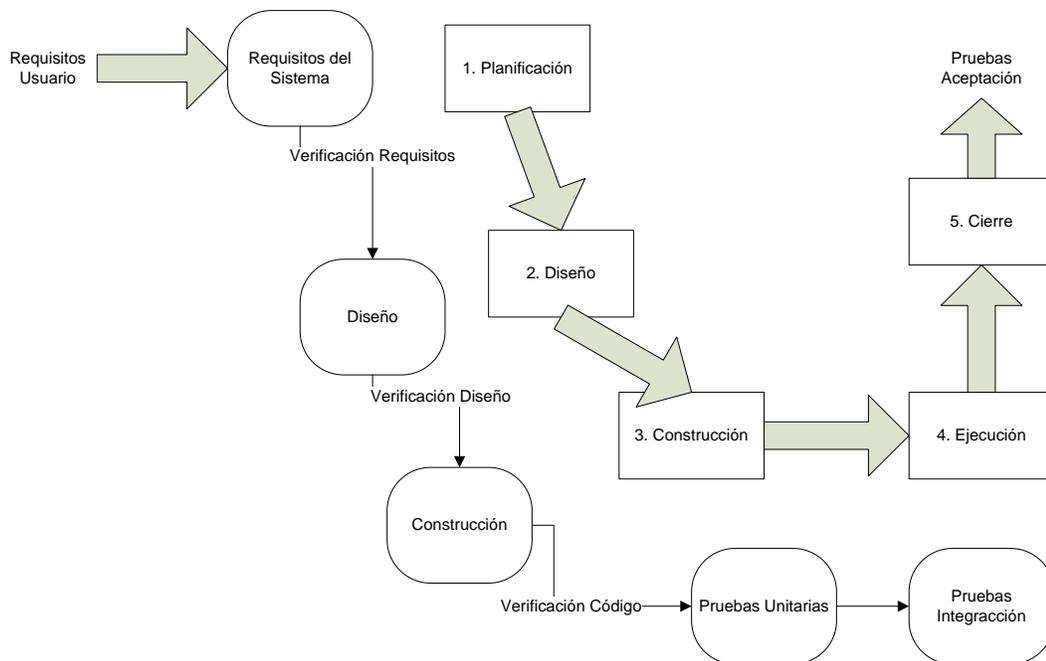
La última fase consiste en el análisis de los datos obtenidos de la ejecución y la redacción de un informe final para los clientes. Los defectos encontrados y los tiempos en las pruebas nos dan como resultado la predicción

de la fiabilidad del software, con la intensidad de los defectos se puede dimensionar el mantenimiento al software.

Con los datos obtenidos podemos crear modelos de crecimiento de fiabilidad del software, con los cuales podemos predecir el número de fallos residuales y podemos decidir cuándo debemos de parar de hacer pruebas.

3.1.2 Proceso paralelo e integrado con ciclo de vida

Figura 27. Diagrama de flujo del proceso paralelo



Este modelo de ejecución de pruebas tiene las mismas etapas que el modelo anterior. La diferencia principal consiste en que se realiza en forma paralela a las etapas del ciclo de vida del producto de software que se va a probar. Los artefactos obtenidos serán los mismos que en el modelo anterior.

En forma general podemos decir que la primera fase del ciclo de vida del software es la colección de los requisitos del sistema, paralela a esta fase se puede realizar la planificación de las pruebas al software.

Las siguientes fase del diseño y construcción de las pruebas se realizan paralelamente a las fases de diseño y construcción del software.

Luego de la construcción del software durante el ciclo de vida se realizan pruebas unitarias y de integración del software, una de estas pruebas se conoce con el nombre de “debuggear”, esta es una actividad que la realiza el desarrollador de software por lo que no se considerará una prueba al software sino que se tomará como una actividad del desarrollo de software. Cuando el producto de software está terminado se procede a la ejecución de la prueba para luego recopilar los datos obtenidos y presentar un informe de la prueba.

3.2 Características tecnológicas

Las herramientas utilizadas en la preparación de las pruebas deben de ser avanzadas, capaz de soportar todas las tecnologías que trabajan los clientes.

Las herramientas de apoyo para las pruebas de software pueden ser:

Herramienta	Descripción
.NET ModelKit Suite	Incluye análisis de datos, visualización y procesamiento de componentes de alta calidad diseñados explícitamente para el Framework .NET
Rational Clear Quest	Ayuda a automatizar y controlar procesos para mejorar la eficacia.

Rational Requisite Pro	Herramienta que facilita la administración y distribución de requerimientos dentro del equipo.
Rational Method Composer	Es una plataforma de manejo flexible de procesos con una librería rica en procesos y herramientas para ayudar a compañías a implementar procesos que garanticen el éxito de proyectos de informática. Su característica principal es que es iterativo.
Rational Test Manager	Es la consola central para realizar pruebas de administración, ejecución y reporte de actividades.
Rational Functional tester	Herramienta para realizar pruebas, automatizada, funcional y de regresión.
Rational Performance tester	Herramienta que permite las pruebas multiusuario y de rendimiento para validar la escalabilidad de las aplicaciones web.
Rational App Scan	Conjunto de soluciones automatizadas de seguridad para aplicaciones Web que escanean y prueban las vulnerabilidades comunes de las aplicaciones Web.
Rational PurifyPlus	Es una solución de análisis en tiempo de ejecución diseñada para ayudar a los desarrolladores a escribir código más fiable y rápido.
Rational Software Architect	Herramienta avanzada de desarrollo dirigido por modelos.

Software gratis

Herramienta	Descripción
OpenSTA	Arquitectura distribuida de pruebas de software diseñada alrededor de CORBA. Tiene la capacidad de realizar pruebas de carga pesada

	de HTTP y HTTPS.
JUnit	Permite realizar la ejecución de clases Java de manera controlada para evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera ^[7] .

Administración de incidentes

Herramienta	Descripción
Mantis Bug Tracker.	Es un sistema gratuito basado en Web para el control de errores.
Sigma7 Help Desk	Gestor de procesos

Redes e Internet

Herramienta	Descripción
Software Firewall (Corta fuegos)	Elemento utilizado en una red de computadoras para controlar las comunicaciones permitiéndolas o prohibiéndolas según las políticas de red definidas en la organización ^[8] .
Software de correo electrónico	Permite la administración de correos electrónicos.

VPN

Herramienta	Descripción
VPN Microsoft	Redes privadas virtuales de Microsoft
VPN Cisco Systems	Redes privadas virtuales de Cisco

Software Adicional

Skype Servicio de telefonía con opción gratuita para usuarios de la red.

3.3 Equipo e instalaciones

Para el equipo y las instalaciones se necesitará lo siguiente:

Hardware

- Impresoras
- Scanner
- Fax

Servidores

- HP Proliant DL380G5

WorkStation

- HP DC 5700

Red

- Switch
- Router
- Cableado
- Wireless

Comunicación

- Teléfono

Enlace a Internet

- Enlace dedicado a Internet

- IP Pública
- Dominio

Infraestructura

- Aire acondicionado
- Espacio

Edificio

- Local

Mobiliario

- Estaciones de trabajo
- Sillas
- Mesas

3.4 Recursos adicionales

Materias primas	Software y documentación
Manejo de inventarios	Tipos de pruebas que el Centro puede realizar
Localización	Cerca de los clientes grandes
Diseño y distribución de planta y oficinas	Diseño de interiores y distribución de localidades con ayuda de un arquitecto
Mano de obra	Consultores Certified Software Test Professional Certified Test Manager

3.5 Procedimientos de mejora continua

Los procedimientos de mejora continua estarán apegados a los estándares de calidad de *Software Engineering – Software Product Quality* (ISO 9126).

Las capacitaciones de los testers y los administradores de pruebas se realizarán por lo menos cada tres años que es lo que dura la certificación.

3.6 Impacto social

La mejora en la calidad de software tendrá varias implicaciones positivas en la industria de software en la región, las cuales se reflejarán principalmente en un aumento en la productividad de los desarrolladores de software. De manera que los productos desarrollados con mejor calidad podrán ser exportados y competir en el mercado internacional.

4. ORGANIZACIÓN

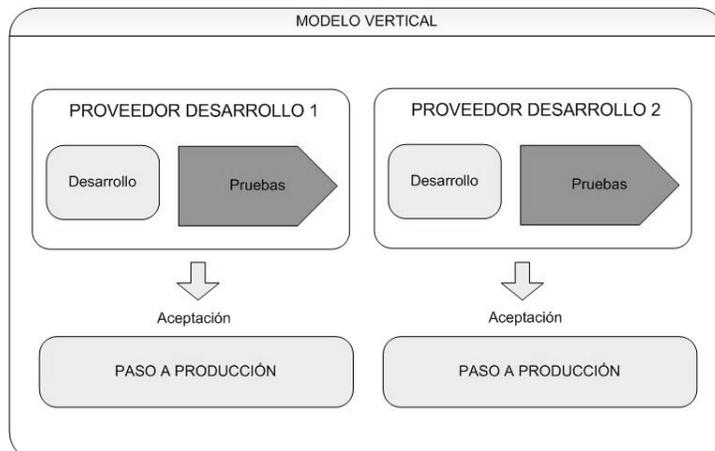
4.1 Modelo de trabajo

Los modelos de trabajo describen la forma de trabajo dentro de la organización

4.1.1 Modelo vertical

En el modelo vertical, las pruebas se trabajan independientemente por cada cliente que pone su software a prueba. Se crea una línea directa de trabajo por cliente desde que se desarrolla el software, pasando por el diseño, construcción, y ejecución de las pruebas para obtener la aceptación del software y colocarlo en producción.

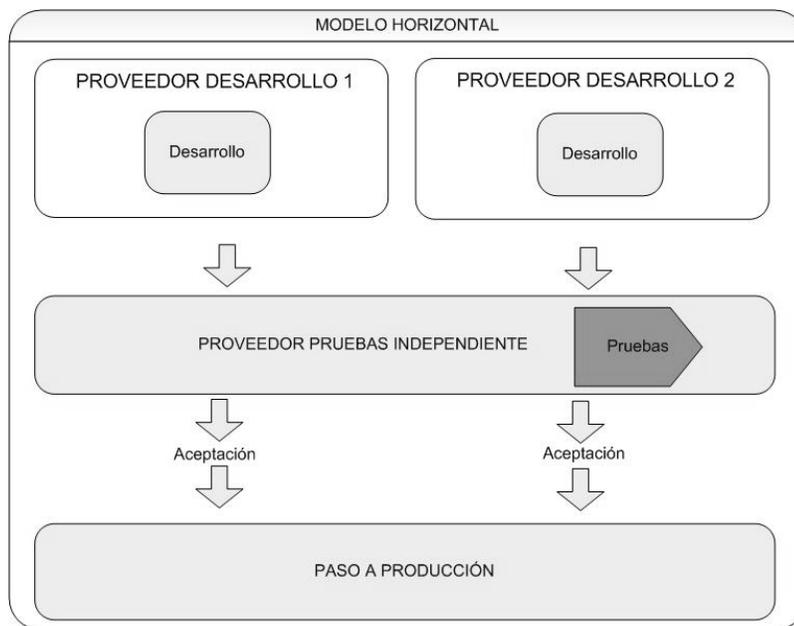
Figura 28. Modelo vertical



4.1.2 Modelo horizontal

El modelo horizontal tiene los mismos pasos que el modelo vertical, en este modelo un equipo de trabajo puede atender a varios clientes a la vez. Inicialmente el equipo de trabajo debe de tener el software terminado con todas sus especificaciones, en la etapa del diseño, construcción y ejecución de las pruebas se incluyen los productos de software que se tengan de los diferentes clientes.

Figura 29. Modelo horizontal

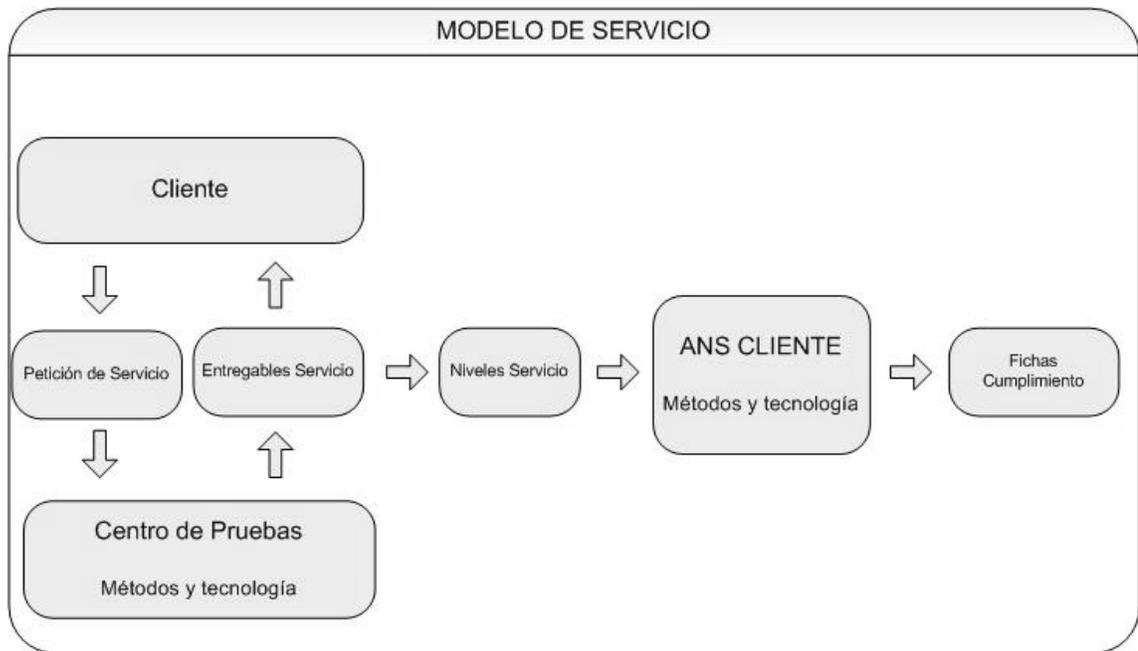


4.1.3 Modelo de servicio

En los modelos anteriores, se puede decir que las pruebas realizadas a los productos de software son pruebas con niveles estándar que maneja el Centro de Pruebas de Software.

En este modelo, la participación del cliente en el proceso de las pruebas es mayor y en conjunto con el personal del centro se definen los acuerdos de niveles de servicio que se proveerán.

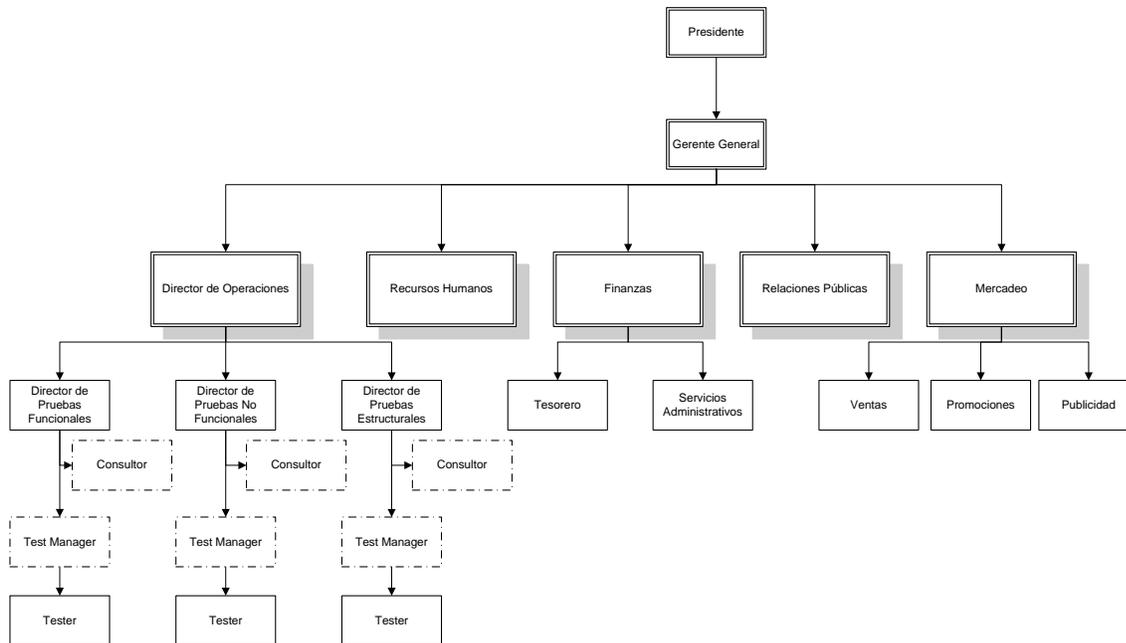
Figura 30. Modelo de servicio



4.2 Estructura organizacional

La estructura organizacional de la empresa presentada a continuación comprende los principales puestos que se desarrollarán en la empresa.

Figura 31. Estructura Organizacional



4.3 Objetivos y funciones del puesto

Puesto	Objetivos
Tester	Ejecutar las pruebas
Test Manager	Diseñar, administrar y presentar resultados de las pruebas
Consultor Externo	Prestar servicios especializados cuando no se cuente con suficiente personal o personal capacitado dentro del Centro de Pruebas.
Consultor Interno	Prestar servicios especializados para el centro de pruebas de auditoría, optimización de procesos, etc. Pero desde un punto de vista externo.
Director de Pruebas Funcionales	Dirigir el modulo de pruebas funcionales para que estas se lleven a cabo de manera correcta

Director de Pruebas No Funcionales	Dirigir el modulo de pruebas no funcionales para que estas se lleven a cabo de manera correcta
Director de Pruebas Estructurales	Dirigir el modulo de pruebas estructurales para que estas se lleven a cabo de manera correcta
Gerente de Operaciones	Velar por que todas las pruebas se ejecuten satisfactoriamente
Gerente de Mercadeo	Diseñar estrategias de mercadeo. Promocionar el Centro de Pruebas
Gerente de Ventas	Ventas de servicios de pruebas
Directivo	Dirección de la empresa para cumplir los objetivos

4.4 Captación y desarrollo del personal

- Definición de perfil
- Entrevista
- Pruebas
- Contratación
- Inducción

4.5 Administración de sueldos y salarios

Las secciones anteriores a este capítulo contienen una estructura organizacional y definiciones de puestos ideales dentro del centro de pruebas de software.

Puesto	Perfil	Sueldo (Mensual)	Número de empleados totales	Total toda la estructura recomendada
<i>Tester</i>	Estudios de nivel medio con título de Bachiller en Computación o estudios universitarios en curso de ingeniería en computación o similar.	US \$500.00	2	US \$1,000.00
<i>Test Manager</i>	Estudios universitarios de administración de proyectos.	US \$1,200.00	1	US \$1,200.00
Consultor Externo	Especialista en el área que se requiera.	US \$2,000.00	1	US \$2,000.00
Consultor Interno	Especialista en administración y auditoría de proyectos de software.	US \$2,000.00	1	US \$2,000.00
Director de Pruebas Funcionales	Estudios universitarios a nivel de licenciatura en ingeniería de computación o similar.	US \$2,200.00	1	US \$2,200.00
	Con conocimientos y experiencia en administración de proyectos.			
	Conocimientos y experiencia en el área de sistemas.			
Director de Pruebas No Funcionales	Estudios universitarios a nivel de licenciatura en ingeniería de computación o similar.	US \$2,200.00	1	US \$2,200.00
	Con conocimientos y experiencia en administración de proyectos.			
	Conocimientos y experiencia en el área de sistemas.			
Director de Pruebas Estructurales	Estudios universitarios a nivel de licenciatura en ingeniería de computación o similar.	US \$2,200.00	1	US \$2,200.00
	Con conocimientos y experiencia en administración de proyectos.			
	Conocimientos y experiencia en el área de computación.			

Gerente de Operaciones	Estudios universitarios a nivel de licenciatura en ingeniería de computación o similar.	US \$2,800.00	1	US \$2,800.00
	Con conocimientos y experiencia en administración de proyectos.			
	Postgrado de administración de proyectos o similar.			
Gerente de Mercadeo	Estudios universitarios a nivel de licenciatura de Mercadeo y publicidad.	US \$2,200.00	1	US \$2,200.00
	Experiencia comprobable en estudios y análisis de mercado en la región.			
Gerente de Ventas	Estudios universitarios a nivel de licenciatura de Mercadeo y publicidad.	US \$2,200.00	1	US \$2,200.00
	Experiencia comprobable en estudios y análisis de mercado en la región.			
Directivo	Estudios universitarios a nivel de licenciatura en ingeniería de computación o similar.	US \$3,200.00	1	US \$3,200.00
	Con conocimientos y experiencia en administración de proyectos.			
	Postgrado de administración de negocios o similar.			
Salarios administrativos y de operaciones extra	limpieza, auxiliares de contabilidad, auxiliares de recursos humanos, vendedores, etc.	US \$700.00	1	US \$700.00
TOTAL				US \$23,900.00

Con el objetivo de crear un presupuesto inicial para abrir el centro de pruebas de software la siguiente tabla incluye los puestos y la cantidad de empleados que se pueden colocar inicialmente.

Puesto	sueldo (Mensual)	Número de empleados para iniciar	Total para iniciar
<i>Tester</i>	US \$500.00	2	US \$1,000.00
<i>Test Manager</i>	US \$1,200.00	1	US \$1,200.00
Gerente de Operaciones	US \$2,800.00	1	US \$2,800.00
Salarios administrativos y de operaciones extra	US \$700.00	1	US \$700.00
TOTAL			US \$5,700.00

5. FINANZAS

5.1 Flujo de efectivo

El siguiente flujo de efectivo muestra una proyección a tres años en dólares de Estados Unidos asumido sin impuestos.

Flujo de Caja (12 meses)

Centro de Pruebas

	Pre-Startup EST	ene-09	feb-09	mar-09	abr-09	may-09	jun-09
Saldo inicial (inicio del mes)	0	120,115	104,683	84,350	64,018	43,685	23,353

ENTRADAS DE OPERACIÓN							
Ventas nacionales							
Ventas internacionales							
Prestamo bancario o de entidades de apoyo	200,000						
TOTAL ENTRADAS	200,000	0	0	0	0	0	0
Total de efectivo disponible (antes de los	200,000	120,115	104,683	84,350	64,018	43,685	23,353

SALIDAS DE							
Compras (Softw are)	16,530						
Compras (equipo de	14,184						
Compras (Administrativo)	1,000						
Salarios		2,800	5,700	5,700	5,700	5,700	5,700
Capacitación en herramientas de softw are, metodología de pruebas y selección de personal	43,000						
Utensilios de oficina	700	700	700	700	700	700	700
mantenimiento	475	475	475	475	475	475	475
Publicidad y promoción			2,000	2,000	2,000	2,000	2,000
Gasolina, viáticos y viajes	269	269	269	269	269	269	269
Contabilidad y legal	505	505	505	505	505	505	505
Renta	1,400	1,400	1,400	1,400	1,400	1,400	1,400
Telefono, Agua, Luz, Internet	1,111	1,111	1,111	1,111	1,111	1,111	1,111
Seguro	56	56	56	56	56	56	56
Otros (varios)	656	656	656	656	656	656	656
SUBTOTAL	79,885	7,971	12,871	12,871	12,871	12,871	12,871
Pago de préstamo		7,461	7,461	7,461	7,461	7,461	7,461
TOTAL PAGADO	79,885	15,432	20,332	20,332	20,332	20,332	20,332
Total de efectivo (fin de mes)	120,115	104,683	84,350	64,018	43,685	23,353	3,021

Flujo de Caja (12 meses)

El año inicia en: ene-09

	jul-09	ago-09	sep-09	oct-09	nov-09	dic-09	Total estimado 2 año 2010	Total estimado 3 año 2010
Saldo inicial (inicio del mes)	3,021	2,688	2,356	2,024	10,991	17,959	24,927	73,528
ENTRADAS DE OPERACIÓN								
Ventas nacionales	20,000	20,000	20,000	20,000	18,000	18,000	180,000	216,000
Ventas internacionales				15,000	15,000	15,000	180,000	216,000
Prestamo bancario o de entidades de apoyo								
TOTAL ENTRADAS	20,000	20,000	20,000	35,000	33,000	33,000	360,000	432,000
Total de efectivo disponible (antes de los	23,021	22,688	22,356	37,024	43,991	50,959	384,927	505,528
SALIDAS DE FINANCIAMIENTO								
Compras (Software)							38,690	
Compras (equipo de oficina y computación)							0	0
Compras (Administrativo)							0	0
Salarios	5,700	5,700	5,700	11,400	11,400	11,400	99,120	286,800
Capacitación en herramientas de software, metodología de pruebas y selección de personal							0	0
Utensilios de oficina	700	700	700	700	700	700	8,400	8,400
mantenimiento	475	475	475	475	475	475	5,701	5,701
Publicidad y promoción	2,000	2,000	2,000	2,000	2,000	2,000	22,000	22,000
Gasolina, viáticos y viajes	269	269	269	269	269	269	3,224	3,224
Contabilidad y legal	505	505	505	505	505	505	6,060	6,060
Renta	1,400	1,400	1,400	1,400	1,400	1,400	16,800	16,800
Telefono, Agua, Luz, Internet	1,111	1,111	1,111	1,111	1,111	1,111	13,330	13,330
Seguro	56	56	56	56	56	56	668	668
Otros (varios)	656	656	656	656	656	656	7,868	7,868
SUBTOTAL	12,871	12,871	12,871	18,571	18,571	18,571	221,863	370,853
Pago de préstamo	7,461	7,461	7,461	7,461	7,461	7,461	89,535	89,535
TOTAL PAGADO	20,332	20,332	20,332	26,032	26,032	26,032	311,398	460,388
Total de efectivo (fin de mes)	2,688	2,356	2,024	10,991	17,959	24,927	73,528	45,140

5.2 Sistema de financiamiento

El financiamiento del proyecto se puede obtener de préstamos a bancos, al BID y a Agexport con el apoyo de entidades gubernamentales.

El monto a financiar es de: US\$200,000.00.

Para iniciar con un proyecto pequeño, el Centro de Pruebas de Software puede iniciar con cuatro personas dedicadas a las pruebas y una a los temas administrativos. El cálculo del monto a financiar tiene como base el apéndice de Gastos estimados al final de este trabajo.

CONCLUSIONES

1. En el mercado guatemalteco existe la oportunidad para el Centro de Pruebas de Software, existe poca utilización de herramientas especializadas para las pruebas y como vimos en el estudio de mercado el control de calidad aplicado actualmente es mínimo. Sin embargo, se tiene bajo puntaje en el interés de adquirir los servicios del centro de pruebas.
2. El aseguramiento de la calidad se da durante todo el desarrollo de los proyectos de software administrando cada una de las pruebas ejecutadas, las herramientas de soporte para la administración de pruebas y el *test* ayudan a esta tarea. Sin embargo la mayoría de las empresas encuestadas en Guatemala no utiliza alguna herramienta de este tipo por lo que el aseguramiento de la calidad para todo el proyecto puede tener resultados pobres.
3. Las herramientas de soporte para las pruebas estáticas son utilizadas para apoyar las pruebas estructurales, las herramientas de soporte para la especificación de pruebas son utilizadas para apoyar las pruebas funcionales, ya que con base a los requerimientos descritos se diseñan las pruebas y se preparan los datos, las herramientas de soporte para la ejecución de pruebas son utilizadas para apoyar las pruebas no funcionales, las herramientas de soporte para el desempeño y monitoreo sirven de apoyo, tanto para las pruebas de confirmación y regresión como para las pruebas no funcionales. Al no utilizar estas herramientas

es más difícil confirmar el buen funcionamiento del sistema cuando se realiza una prueba.

4. Errar es de humanos y esto es particularmente cierto en los sistemas de software, por lo que la ayuda de las herramientas para las pruebas a estos sistemas es necesaria para minimizar los errores y optimizar la calidad que se entrega a los clientes.
5. Consultoría en la ejecución de pruebas. Debido a que las herramientas son poco utilizadas en la ejecución de las pruebas, es una buena oportunidad dar consultoría en la ejecución de cada una, de acuerdo a los estándares existentes para cada una de las pruebas
6. Pruebas funcionales. Estas pruebas son normalmente utilizadas sin herramienta de apoyo, por lo que proporcionar este servicio adecuadamente con los estándares de calidad sería de beneficio, tanto para la empresa desarrolladora como para el cliente.
7. Pruebas no funcionales. Estas pruebas son aplicadas a las características del software, estas son normalmente aplicadas sin herramienta de apoyo, por lo que al proporcionar este servicio por parte del centro de pruebas contribuiría al desarrollo de software de calidad en Guatemala.
8. Pruebas estructurales. Estas pruebas son normalmente aplicadas sin herramientas de apoyo, por lo que el servicio de estas pruebas por parte del Centro de Pruebas contribuiría en la estandarización de técnicas de desarrollo, y la constante actualización en las arquitecturas de software utilizadas. Las pruebas relacionadas a cambios, principalmente

generadas por defectos, se reducirían debido a que las pruebas anteriores aseguran la calidad del software desarrollado.

9. El 100% de las empresas asociadas a SOFEX proveen servicios de diseño de software y el 60% provee soluciones a la medida. El 80% de estas empresas demandan capacitaciones de Administración de Proyectos y el 60% demanda capacitaciones de Control de Calidad. El Centro de Pruebas de Software tiene la oportunidad de cubrir esa demanda.

10. El modelo de ejecución de las pruebas más adecuado es el de tener acceso a los servidores de los clientes desde las oficinas locales del centro de pruebas, mediante una conexión segura a través de Internet.

RECOMENDACIONES

1. En los resultados obtenidos, se puede observar cómo las empresas de software en Guatemala se mantienen en constante movimiento. Por lo que se puede aprovechar esta habilidad de dichas empresas para agregarles un sello de calidad no solo a nivel nacional sino internacional.
2. Crear no solo una empresa que dé control de calidad, por medio de pruebas, sino un sello que tenga peso ante las distintas empresas a nivel internacional. Para esto se puede realizar una alianza estratégica con universidades del país o con entidades del gobierno.
3. La iniciativa del Centro de Pruebas de Software puede ser otro de los servicios que Guatemala, puede ofrecer, como lo hace actualmente con los centros de llamadas.
4. Presentar proyectos a entidades que puedan financiar proyectos de este tipo, manteniendo la métrica de generación de empleo para el país es algo que se puede explotar y convertir a Guatemala en un competidor importante en este mercado.
5. Orientar la empresa a las pruebas destacadas y no destacadas en los resultados de la encuesta. El hecho que ya realicen las pruebas muestra interés y necesidad de hacerlo. Y el que no las hagan puede mostrar una necesidad por el hecho de encontrar esa deficiencia en el software.

6. Realizar campañas publicitarias para motivar y alentar a los clientes potenciales a adquirir el servicio de pruebas de un ente externo, mostrándoles las ventajas que esto podría traerles a su empresa. Construir una campaña de concientización y capacitación con el apoyo de SOFEX ofreciendo charlas y seminarios a los clientes potenciales.
7. Adoptar modelos de calidad como el ISO 9126, porque es un estándar internacional para la evaluación de software que provee un entorno para que las organizaciones definan un modelo de calidad para el producto de software.
8. Apoyarse en las herramientas diseñadas para cada tipo de prueba. Las herramientas utilizadas en el Centro de Pruebas de Software deben de ser elegidas de acuerdo a la tecnología utilizada para el desarrollo de las aplicaciones. De acuerdo con los resultados de SOFEX el 87% de sus asociados utiliza la tecnología .NET para el desarrollo de sus aplicaciones, 67% utiliza Visual Basic y 53% Java. En cuanto a base de datos el 93% de los asociados utiliza SQL Server, 67% utiliza Oracle y 47% My SQL. Las tecnologías anteriormente mencionadas no son las únicas, pero el Centro de Pruebas debe de ser capaz de construir cualquier prueba sobre cualquiera de estas tecnologías.
9. En el mercado guatemalteco existe la oportunidad para el Centro de Pruebas de Software, el bajo puntaje en el interés de adquirir los servicios del centro de pruebas exigirá crear una estrategia de ventas fuera del país, para lograr tener un negocio rentable.

10. Iniciar a realizar las pruebas de software con un enfoque internacional para el mercado local, al adquirir experiencia, promocionar internacionalmente para tener contactos que hagan el centro rentable.

REFERENCIAS

1. <http://www.rational.com.ar/cmmi/cmmiesp.html>
Fecha consultada: Septiembre 2008.
2. <http://es.wikipedia.org/wiki/CMMI>
Fecha consultada: Septiembre 2008.
3. <http://es.wikipedia.org/wiki/Moprosoft>
Fecha consultada: Septiembre 2008.
4. <http://www.amcis.org.mx/>
Fecha consultada: Enero 2008.
5. <http://www.export.com.gt/Portal/Home.aspx?secid=713>
Fecha consultada: Septiembre 2008.
6. http://www.trabajo.com.mx/las_4_p_de_la_mercadotecnia_plaza.htm
Fecha consultada: Enero 2008.
7. <http://es.wikipedia.org/wiki/JUnit>
Fecha consultada: Septiembre 2008.
8. http://es.wikipedia.org/wiki/Imagen:Gateway_firewall.svg
Fecha consultada: Septiembre 2008.

BIBLIOGRAFÍA

1. Centro de Ensayos de Software, CES.
<http://www.ces.com.uy>
Jornada de *Testing* en Vivo – Centro de Ensayos de Software
Montevideo, 26 de Abril de 2006
Estado del Arte de las Pruebas de Software
2. International Software Testing Qualifications Board
<http://www.istqb.org/>
3. International Institute for Software Testing
<http://www.iist.org/>
4. Comisión de Software, SOFEX
<http://www.export.com.gt/Portal/Home.aspx?sub=SOFEX>
5. deGuate.com
<http://www.deguate.com>
6. Software Testing
Autor Paul C. Jorgensen
Referencia: Búsqueda de libros de Google.

7. International Institute for Software Testing
www.iist.org

8. Gaiasur
<http://planeta.gaiasur.com.ar/>

9. Instituto Tecnológico de Informática
<http://www.iti.upv.es/groups/squac>

10. Wikipedia
<http://es.wikipedia.org>

11. Vocabulario de ordenadores e internet
<http://jamillan.com/>

12. El emprendedor de éxito. Guía de planes de Negocios. Segunda edición.
Rafael Eduardo Alcaraz Rodríguez
Instituto Tecnológico y de Estudios Superiores de Monterrey. Campus
Monterrey. McGraw-Hill

APÉNDICES

Encuesta I

http://www.encuestafacil.com/RespWeb/Cuestionario.aspx?EID=23168&MT=X&M5J=NO_COPIAR_ESTES_LINK

 encuestafacil.com

Pruebas de Software

[Abandonar->](#)

1.- Introducción

Con el fin de conocer el nivel de aseguramiento de la calidad que se tiene en el software guatemalteco necesitamos su ayuda respondiendo unas preguntas que le tomará únicamente cinco minutos de su tiempo.

Pag. 1 / 4

[Siguiete->](#)

Crea tus propias encuestas en www.encuestafacil.com

Abandonar->

2.- Características

Esta sección explora su punto de vista de las pruebas de software

***1. Dentro de la organización se realizan pruebas al software para:** *Es imprescindible que conteste a esta pregunta*

- Describir, con ejemplos, la forma en la que un defecto en el software puede dañar a una persona, al ambiente o a una compañía
- Distinguir la causa principal de un defecto y sus efectos
- Dar razones del porque las pruebas son necesarias dando ejemplos
- Describir porque las pruebas forman parte de la aseguración de la calidad y dar ejemplos de cómo las pruebas contribuye a una mejor calidad
- Recordar los términos de error, defectos, fallas y términos correspondientes de error y bugs

***2. Porcentaje del desarrollo del proyecto dedicado a pruebas de software**

- 0 - 5 %
- 6 - 10 %
- 11 - 20 %
- 21 - 50 %
- Durante todo el desarrollo del proyecto

***3. ¿Participa en la planeación, diseño o ejecución de las pruebas al software?**

- Sí
- No

Pag. 2 / 4

<-Anterior Siguiente->

Crea tus propias encuestas en www.encuestafacil.com

Abandonar->

3.- Tipos de pruebas

En esta sección conoceremos los tipos de pruebas que existen y el nivel de aplicación en su organización

***4. ¿Para todos los proyectos realizan pruebas funcionales? Basadas en las funciones (lo que el sistema hace) y características (descritas en documentos y comprendidas por los testers). Considera el comportamiento externo del sistema (pruebas de caja negra).**

Nunca Casi nunca Normalmente Casi siempre Siempre

***5. ¿Para todos los proyectos realizan pruebas a las características del software? Son las pruebas de cómo trabaja el sistema. Estas pruebas requieren de características medibles de los sistemas y el software que pueden ser cuantificables en una escala variable, como por ejemplo los tiempos de respuesta para las pruebas de rendimiento.**

Nunca Casi nunca Normalmente Casi siempre Siempre

***6. ¿Para todos los proyectos realizan pruebas a la estructura/arquitectura del software? Pruebas estructurales (pruebas de caja blanca) pueden realizarse en todos los niveles de pruebas. Especialmente en las pruebas de integración de componentes.**

Nunca Casi nunca Normalmente Casi siempre Siempre

***7. ¿Realizan pruebas relacionadas a cambios? Cuando se detecta un defecto y es arreglado, el software debe ser probado nuevamente para confirmar que el defecto haya sido removido.**

Nunca Casi nunca Normalmente Casi siempre Siempre

***8. ¿El equipo encargado de la aseguración de la calidad administra las pruebas al software?**

Si
 No

Pag. 3 / 4

<-Anterior

Siguiente->

Crea tus propias encuestas en www.encuestafacil.com

Pruebas de Software Abandonar->

4.- Tipos de herramientas

En esta sección conoceremos los tipos de herramientas que nos ayudan a realizar las pruebas al software

***9. ¿Utilizan herramientas de soporte para la administración de las pruebas? Administración de los requerimientos, Administración de los incidentes, Administración de la configuración**

No
 Sí (Mencione la mas utilizada)

***10. ¿Utilizan herramientas de soporte para las pruebas estáticas? Análisis estático para apoyar los estándares de desarrollo, el análisis de las estructuras y dependencias y poder ayudar al entendimiento del código. Herramientas de modelado, las cuales validan los modelos del software.**

No
 Sí (Mencione la mas utilizada)

***11. ¿Utilizan herramientas de soporte para la especificación de las pruebas? Herramientas para el diseño de pruebas, herramientas para la preparación de los datos de prueba**

No
 Sí (Mencione la mas utilizada)

***12. ¿Utilizan herramientas de soporte para la ejecución de las pruebas y el log? Herramientas para la ejecución de pruebas, herramientas que simulan el ambiente en el cual el objeto a probar se ejecutará, comparadores de pruebas y herramientas de seguridad**

No
 Sí (Mencione la mas utilizada)

***13. ¿Utilizan herramientas de soporte para desempeño y monitoreo? Herramientas de análisis dinámico, herramientas para pruebas de estrés y herramientas de monitoreo**

No
 Sí (Mencione la mas utilizada)

Pag. 4 / 4 <-Anterior Fin->

Encuesta II

Centro de Pruebas de Software

Entrevista

Guatemala está en vías de exportar servicios diferenciados, entre ellos el desarrollo de software, para ello se está investigando la factibilidad de colocar un Centro de Control de Calidad de Software en el país, solicitamos su ayuda para obtener información y evaluar este estudio.

Parte I

Conociendo a la empresa

1. Cantidad de clientes:

	1 – 5
	6 – 10
	11 – 20
	20 – 50
	50 – 100
	100 – 500
	500 – 1000
	> 1000

2. Número de empleados:

	1 – 10
	11 – 50
	51 – 200
	200 – 500
	500 – 1000
	> 1000

3. ¿Contrata actualmente servicios externos de control de calidad y pruebas de software?

SÍ __

(Siga en 4)

NO __

(Pase a 5)

4. ¿De qué proveedor contrata servicios de control de calidad y pruebas de software?

Parte II

Acerca del desarrollo.

5. Número de programadores:

	1 – 5
	6 – 10
	11 – 20
	21 – 30
	31 – 50
	> 50

6. Cantidad de proyectos nuevos de desarrollo de software al año:

7. Cantidad de personas por proyecto de desarrollo de software

	1 – 2
	3 – 5
	6 – 10
	> 10

8. Cantidad de productos de software para venta que poseen:

9. Realizan desarrollo de software a la medida:

SÍ _____ NO _____

(Pase a 12) (Siga en 10)

10. Edad del producto líder

a. Años: _____

11. Versión del producto líder

a. Versión del producto: _____

12. Nivel de implementación en su empresa de las siguientes prácticas.

Donde 1 es no aplica y 5 es aplica completamente

Software Configuration Management: Trata y controla la elaboración de código fuente por varios desarrolladores simultáneamente, el seguimiento del estado de las versiones y sus cambios y la conducción de la integración de las partes del software en un solo					
Administración de proyectos: Aplicación del enfoque de sistemas para la administración de tareas tecnológicas complejas o de proyectos cuyos objetivos se establecen explícitamente en términos de tiempo, costos y parámetros de realización					
Administración de los requerimientos: Comprende las actividades relacionadas con la definición, clasificación, asignación, seguimiento y control de los requerimientos durante todo el ciclo de vida de desarrollo de software					
Control de calidad: Estrategia para asegurar el mejoramiento continuo de la calidad					

13. Interés en adquirir los servicios de control de calidad de software de una empresa externa para productos nuevos de la empresa

Nada interesado					Muy interesado				
1	2	3	4	5	6	7	8	9	10

Parte III

Acerca del Control de Calidad

14. Porcentaje de errores reportados al departamento de desarrollo de software en las solicitudes de soporte

_____ %

15. De la siguiente lista de ensayos de plataforma, asigne el nivel de interés en adquirirlos por parte de un Centro de Pruebas de Software. Donde 1 es nada interesado y 5 es muy interesado

Test de carga: Estudio del desempeño de productos bajo distintos niveles de carga esperables para lo cual fue diseñado					
Test de estrés: Estudio del desempeño de productos ante situaciones de carga para los cuales no fue diseñado					
Test de escalabilidad: Estudio del desempeño del producto respecto a cambios en los recursos disponibles					
Test de configuración: Estudio del desempeño del producto ante cambios de plataforma base, tanto de Hardware como de Software					

16. De la siguiente lista de servicios de pruebas, asigne el nivel de interés en adquirirlos por parte de un Centro de Pruebas de Software. Donde 1 es nada interesado y 5 es muy interesado

Test funcional: Estudio del comportamiento de las funcionalidades y ciclos de funcionalidades del producto					
Test de integración: Estudio del comportamiento de la integración de distintos módulos o componentes del producto					
Test de usabilidad: Estudio de la facilidad de uso del producto para los usuarios meta del sistema bajo las condiciones de trabajo previstas					
Test de seguridad: Estudio del grado de protección que ofrece el producto contra ataques externos y accesos no autorizados hacia el sistema y sus datos					
Test de robustez: Estudio del comportamiento del sistema en casos que no fueron previstos en la especificación del mismo					
Test de compatibilidad: Estudio del comportamiento del producto cuando éste interactúa con otros sistemas					

17. Porcentaje del presupuesto por proyecto asignado a control de calidad
 _____ %

18. ¿Cómo considera el control de calidad aplicado actualmente?

	No aplica
	Mínimo
	Regular
	Adecuado
	Excelente

19. Nombre algunas de las empresas que provean servicios de pruebas de software en su región

Gastos estimados

COMPRA DE SOFTWARE	
	Precio
.NET ModelKit Suite	US \$ 1,600.00
Clear Quest	US \$ 7,240.00
Requisite Pro	US \$ 2,250.00
Method Composer	US \$ 419.00
Functional tester	US \$ 5,440.00
Performance tester	US \$ 1,600.00
Rational App Scan	US \$32,500.00
PurifyPlus	US \$ 1,450.00
Software Architect	US \$ 3,140.00

COMPRA DE MAQUINARIA Y EQUIPO	
Server HP Proliant	US \$ -
CPU HP DC 5700	US \$ 3,436.00
Monitor LCD 15"	US \$ 748.00
Seguridad y Respaldo	US \$ 3,000.00
Red y Conectividad	US \$ 10,000.00

COMPRAS ADMINISTRATIVAS	
Equipamiento Administrativo	US \$ 1,000.00
Total	US \$ 1,000.00

UTENSILIOS DE OFICINA	
Utensilios de oficina	US \$ 500.00
Complementos y artículos de limpieza	US \$ 200.00
Total	US \$ 700.00

Captación y capacitación de personal		
Desarrollo de perfiles	US \$	7,000.00
Contratación de personal	US \$	6,000.00
Curso Inicial Foundation		
Negociación de Capacitación on-site	US \$	1,000.00
Viáticos Instructor a Guate	US \$	3,500.00
Infraestructura Curso y Materiales	US \$	2,500.00
Examen Foundation	US \$	1,000.00
Otros cursos de soporte		
A tomar en USA o España	US \$	16,000.00
Viáticos 2 personas	US \$	6,000.00
Total	US \$	43,000.00

Otros Gastos		
Gastos internos mensuales		
Seguridad	US \$	55.70
Viáticos, gastos de personal	US \$	600.00
Total		\$655.70