

03063
15



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

**“PRUEBAS, AMBIENTE Y DISTRIBUCIÓN
BASADOS EN EL PROCESO UNIFICADO
(UN CASO PRÁCTICO)”**

T E S I S

QUE PARA OBTENER EL GRADO DE:

**MAESTRO EN INGENIERÍA
(COMPUTACIÓN)**

P R E S E N T A:

JOSÉ ANTONIO SALAZAR CARMONA

DIRECTORA DE LA TESIS: M. EN C. MARÍA GUADALUPE E.
IBARGÜENGOITIA GONZÁLEZ.

TESIS CON
FALLA DE ORIGEN

MÉXICO, D.F.

2003.

À



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

A Dios, que me ha dado magnificas facultades.

A mis padres, Ignacio y Cristina que me dieron una valiosa educación de perseverancia.

A mis hermanos Silvia, Emilio, Carlos, Alfredo, Jorge y Jacqueline por su apoyo incondicional.

A mi esposa Araceli por alentarme y creer en mi.

y

A mi hijo Leonardo, por animarme con su entusiasmo.

TESIS CON
FALLA DE ORIGEN

INDICE

Introducción 1

Capítulo 1: Conceptos fundamentales 3

Generalidades del modelado de procesos de software 3

Proceso Unificado 4

Características del Proceso Unificado 4

Flujos de trabajo 7

Proceso Unificado de Rational 8

Diagramas de UML 9

Instituto de Investigaciones Bibliográficas 11

Departamento de Informática 11

Capítulo 2 : Planteamiento de la adaptación 13

Objetivos 14

Análisis para hacer la adaptación 14

Valoración del departamento de informática 15

Capítulo 3: Flujo de trabajo ambiente 17

Actividades 17

Conocer el ambiente para el proyecto 21

Preparar el ambiente para un ciclo 24

Roles 26

Especialista en herramientas 26

Artefactos 27

Estándar del proceso de desarrollo de sistemas del IIB (PDSIIB) 27

Estándar para modificar el proceso 30

Comité responsable del PDSIIB 32

Solicitud de cambio al PDSIIB 33

Estándar de herramientas 34

Herramientas a comprar 37

Configuración de herramientas 39

¡TENIE CON
FECHA DE ORIGEN

Capítulo 4: Flujo de trabajo distribución 41

Actividades 41

- Hacer plan de distribución 45
- Hacer manuales 48
- Realizar pruebas de aceptación en desarrollo 51
- Producir entregable 54
- Realizar pruebas de aceptación en sitio 56

Roles 59

- Administrador de distribución 59
- Escritor técnico 59

Artefactos 60

- Estándar de distribución 60
- Estándar para los manuales del sistema 64
- Plan de distribución 65
- Invitación de pruebas formales 69
- Aceptación de pruebas en desarrollo 71
- Informe de entregable 73
- Aceptación de pruebas en sitio 76

Capítulo 5: Flujo de trabajo pruebas 79

Actividades 79

- Hacer el plan de pruebas 84
- Hacer el diseño de pruebas 85
- Ejecutar pruebas 87
- Evaluar las pruebas 89

Roles 92

- Diseñador de pruebas 92
- Probador 92

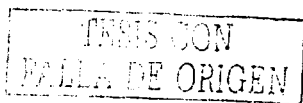
Artefactos 93

- Estándar de pruebas 93
- Plan de pruebas 98
- Caso de prueba 101
- Reporte de evaluación de pruebas 103

Capítulo 6: Herramienta para el PDSIIB 105

- Planteamiento de la herramienta 105
- Objetivos 105
- Uso de la herramienta 106

Conclusiones 109



Apéndices 113

Apéndice A: Flujo de trabajo ambiente de acuerdo al Proceso Unificado de Rational 113

Actividades 113

- Preparar el ambiente para el proyecto 113
- Preparar el ambiente para una iteración 127
- Preparar las guías del proyecto para una iteración 129
- Apoyar el ambiente durante la iteración 129

Apéndice B: Flujo de trabajo distribución de acuerdo al Proceso Unificado de Rational 131

Actividades 131

- Desarrollar el plan de distribución 131
- Desarrollar el material de apoyo 133
- Administrar las pruebas de aceptación 135
- Producir la unidad de distribución 137
- Administrar pruebas beta 139
- Empaquetar el producto 140
- Proporcionar acceso a un sitio para la descarga 142

Apéndice C: Flujo de trabajo pruebas de acuerdo al Proceso Unificado de Rational 143

Actividades 143

- Hacer el plan de pruebas 143
- Diseñar las pruebas 149
- Implementar las pruebas 154
- Ejecutar las pruebas de integración 156
- Ejecutar las pruebas de sistema 158
- Evaluar las pruebas 160

Apéndice D: Ejemplo de llenado de los artefactos del Flujo de Trabajo Ambiente 163

- Herramientas a comprar 164
- Configuración de herramientas 165

Apéndice E: Ejemplo de llenado de los artefactos del Flujo de Trabajo Distribución 167

- Plan de distribución 168
- Invitación de pruebas formales 171
- Aceptación de pruebas en desarrollo 172
- Aceptación de pruebas en sitio 173
- Informe de entregable 174

TESIS CON
FALLA DE ORIGEN

**Apéndice F: Ejemplo de llenado de los
artefactos del Flujo de Trabajo Pruebas 175**

Plan de pruebas 176

Casos de prueba 178

Reporte de evaluación de pruebas 182

Bibliografía 183

Introducción

Construir sistemas de software en la época actual implica poner atención a los problemas ocultos que afectan a las organizaciones de desarrollo de software. Algunos de estos problemas se deben a: la sofisticación, la variedad de requerimientos, la interrelación, el volumen de los datos, la diversidad de la tecnología, la obsolescencia, los riesgos, el tiempo y el costo. Son muchas las cosas que se tienen que administrar, cada problema tiene varias formas de solución, sin embargo, se han hecho presentes los procesos de software, los cuales consideran la idea de tratar al software como un producto, como lo hacen las otras áreas de ingeniería, pero con alguna connotación distinta. Los sistemas deben de contar con la capacidad de establecer una arquitectura abierta, facultad para brindar reusabilidad de componentes, utilización de estándares, creación de diseños modulares entendibles y fáciles de mantener.

La creación de un proceso de software es muy parecida a la creación de un producto de software, por lo que hay que tener cuidado en no incurrir en los vicios que produjeron la crisis del software, podría surgir nuevamente la crisis de los procesos de software. Esto es porque, el software evoluciona y el proceso de software también, el software se hace para requerimientos específicos y el proceso de software para organizaciones específicas, el software se debe documentar y el proceso de software también, el software se debe probar y el proceso de software también, al software se le debe dar mantenimiento y al proceso de software también.

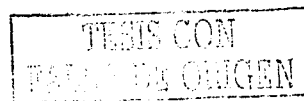
Se resalta la importancia de la implementación de procesos, pero sobre todo que estos procesos se utilicen y después sean modificados hasta que alcancen su madurez.

El Proceso Unificado fue escogido como base para hacer la adaptación debido a que es un marco general, se usa cada día más, contempla las experiencias de personas e instituciones, manifiesta algunas de las mejores prácticas de desarrollo de software y porque es adaptable.

En este trabajo se adaptan tres flujos de trabajo, pruebas, ambiente y distribución, la razón de que únicamente sean tres flujos de trabajo es porque son muchas actividades, artefactos y roles que se deben estudiar para obtener el proceso completo y una sola persona no podría abarcar todos los flujos que contiene el Proceso Unificado. Lo cautivante de este tema de investigación sobre la adaptación, es que está integrado por un equipo de trabajo para obtener el proceso completo, disponible para su consulta en esta tesis y en otras dos más tituladas "La implementación y administración del proyecto, cambios y configuración del Proceso Unificado adaptados a un caso práctico"[Ramírez], y "Adaptación del Proceso Unificado en los flujos de trabajo de requerimientos, análisis y diseño: caso práctico"[Sánchez], juntos conforman la adaptación del proceso de desarrollo de sistemas del Instituto de Investigaciones Bibliográficas de la UNAM.

Como resultado concreto se crea una herramienta que a través de diagramas de actividad describe los pasos, roles y artefactos que se utilizarán en el desarrollo de sistemas.

En el capítulo uno se mencionan los conceptos fundamentales del modelado de procesos, del Proceso Unificado, las características principales, los flujos de trabajo y las fases. Se explican los conceptos del Lenguaje Unificado de Modelado (UML). Además se incluye una breve historia del Instituto de Investigaciones Bibliográficas y sobre las funciones del Departamento de Informática.



TESIS CON
FALLA DE ORIGEN

En el capítulo dos se hace el planteamiento de la adaptación y sus objetivos, el análisis que se hizo para la adaptación y el resultado de la valoración del departamento de informática que después servirá para adaptar cada flujo de trabajo.

En el capítulo tres se explica la adaptación realizada sobre el flujo de trabajo ambiente de acuerdo a como lo indica el Proceso Unificado de Rational.

En el capítulo cuatro se explica la adaptación realizada sobre el flujo de trabajo distribución de acuerdo a como lo indica el Proceso Unificado de Rational.

En el capítulo cinco se explica la adaptación realizada sobre el flujo de trabajo pruebas de acuerdo a como lo indica el Proceso Unificado de Rational.

En el capítulo seis se hace el planteamiento de una herramienta que contenga el proceso de desarrollo de sistemas que resultó de la adaptación, el propósito de crear esta herramienta es tener una guía para realizar las actividades del proceso.

En las conclusiones se menciona la experiencia obtenida de esta particular adaptación.

En la parte de apéndices se incluyen las descripciones completas de las actividades de los tres flujos de trabajo tomados del Proceso Unificado de Rational y se muestran ejemplos del llenado de los artefactos.

Capítulo 1

Conceptos fundamentales

Generalidades del modelado de procesos de software

Desde que aparecieron las tecnologías de información han tenido un éxito aceptable debido a que han resuelto los problemas que han surgido desde tiempos pasados hasta los actuales, ya que el volumen de los datos y la complejidad de las relaciones entre éstos ha ido creciendo considerablemente y como consecuencia a aparecido el desarrollo de software para manipular información.

El trabajo requerido para construir, mantener y utilizar sistemas requiere de un diseño organizacional, una guía disciplinada y herramientas de apoyo. La ingeniería de procesos se involucra para relacionar todos estos elementos.

La ingeniería de procesos es un área para estudiar los procesos organizacionales a través de técnicas para el análisis, diseño y evolución de los procesos que son establecidos a partir del modelado.

El modelado de procesos permite entender el proceso, evaluarlo, seleccionar entre diferentes alternativas de procesos, diseñar nuevos procesos, mejorar algunos ya existentes, ajustar o reusar partes de otros procesos y llevar a cabo dichos procesos. Para que un modelo de procesos sea calificado como completo debe considerar con claridad qué se va hacer, quién lo va hacer, cuándo, dónde y cómo. Un proceso tiene actividades organizadas en diferentes fases del ciclo de vida del desarrollo de software.

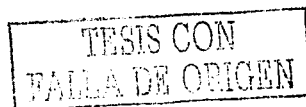
Los objetivos del modelado de procesos de software se encaminan a ofrecer soporte a la administración de proyectos, facilitar la comunicación entre grupos de personas a través de estándares, contribuir a mejorar procesos existentes, brindar guías para establecer descripciones y apoyar la ejecución de actividades.

Las ventajas del modelado de procesos sobresalen por las características que ofrece para representar agentes que ejecutan actividades, interacciones entre roles, manipulación de información contenida en los artefactos y la naturaleza innata para realizar mejoras.

Los modelos son representaciones abstractas de la realidad que excluyen los detalles. El propósito de un modelo es reducir la complejidad del comportamiento de un fenómeno eliminando detalles que no tienen influencia. De esta manera un modelo destaca lo que su creador considera importante para la comprensión del fenómeno, la selección de límites depende de los usos en los cuales el modelo será aplicado. El modelado de procesos tiene implícitos varios elementos que se enuncian a continuación.

Actor es alguien o algo externo al sistema que interactúa con éste.

Agente es un actor humano o máquina que lleva a cabo actividades de un proceso.



Rol es la definición del comportamiento y responsabilidades de un individuo o equipo de trabajo dentro del contexto de una organización que desarrolla software. El comportamiento específico es nombrado de acuerdo al contexto particular. Los agentes son a los que se les asignan roles.

Artefacto es una pieza de información que es generada, modificada y utilizada por un proceso definiendo un área de responsabilidad. Un artefacto puede ser un modelo, un elemento del modelo, un documento o software. Tiene como sinónimo la palabra producto.

Con los elementos anteriores, se define **proceso de software**, como un conjunto de pasos ordenados que serán ejecutados por agentes responsables de un rol utilizando artefactos para obtener un producto de software.

La investigación de los procesos de software ha sido dirigida a la exploración de la forma y contenido de los procesos, cómo afectan las características del proceso a los productos obtenidos, cómo los procesos pueden ser explotados por las organizaciones que desarrollan software y cómo los procesos pueden ser soportados por herramientas y tecnología.

Los procesos de software son procesos complejos que son llevados a cabo por equipos de personas que utilizan métodos y herramientas que influyen en la calidad del software, como el costo, el riesgo y el tiempo.

La reingeniería de procesos de negocio se utiliza principalmente para reestructurar los diseños de procesos de negocio de una organización con el propósito de hacer, en la medida de lo posible, procesos eficientes.

Los ambientes de desarrollo de procesos de software emplean sistemas automáticos que integran el trabajo de administración y del personal técnico involucrado para obtener un proceso de desarrollo de software disciplinado y delimitado.

Proceso Unificado

Características del Proceso Unificado

Las expectativas a las que se debe acoplar el desarrollo de un sistema de software hacen que sea conveniente contar con un proceso de ingeniería donde se defina con claridad quién debe de hacer qué, cuándo y cómo. Las necesidades son inmensas, los usuarios son más exigentes, demandan aplicaciones grandes y complejas e incluso quieren que se desarrollen más rápido, la dificultad estriba en la coordinación de diversas actividades que deben realizarse en los proyectos.

“El Proceso Unificado es un proceso de desarrollo de software. Un Proceso de desarrollo es el conjunto de actividades necesarias para transformar los requisitos de usuario en un sistema software.”[Jacobson]

El Proceso Unificado está guiado por casos de uso, dirigido por la arquitectura y orientado por iteraciones. Se lleva a cabo a través de varios ciclos, en cada ciclo se hacen entregas parciales del sistema. Cada ciclo consta de cuatro fases: inicio, elaboración, construcción y transición. Cada fase se subdivide a su vez en iteraciones que involucran a los flujos de trabajo para que a partir de ellos se realicen las actividades del proyecto.

Las actividades de desarrollo las realizan personas, por lo que es preciso que el proceso sea del conocimiento común de todas ellas. La administración y organización de un

proyecto afectan a los individuos responsables, por lo que, la planeación del proyecto, la viabilidad, la gestión de riesgos, la coordinación de los equipos de desarrollo y la comprensión del proyecto tienen que ser manejadas como ideas básicas para conseguir resultados fructíferos.

Las personas desempeñan roles que les son asignados y por eso es conveniente saber que la productividad se obtiene de forma natural cuando tienen habilidades acordes a las actividades que van a desempeñar. Algunas habilidades pueden conseguirse mediante formación y algunas otras se obtienen a través de la experiencia.

El desarrollo del sistema se inicia con la construcción de modelos, utilizando "distintos modelos para describir todas las perspectivas del sistema." [Jacobson] Tener un proceso universal bien definido ayuda a que todas las personas comprendan lo que tienen que hacer, todos pueden entender lo que los otros están haciendo. Para toda persona resulta fácil comprender lo que se está haciendo en un momento determinado gracias al modelo de la arquitectura. Cuando existe la necesidad de cambiar personas a otros proyectos no habrá incertidumbre, pues no tendrán que aprender una nueva forma de definir o comprender especificaciones de procesos.

Casos de uso

La administración de sistemas de calidad se consigue cuando las necesidades de los clientes se incorporan, los casos de uso proporcionan un medio sistemático de obtener requerimientos funcionales dirigiendo todo el proceso de desarrollo debido a que la mayoría de las actividades se llevan a cabo partiendo de ellos.

Para cada caso de uso se identifican las tareas a realizar, por lo que se puede estimar el esfuerzo y tiempo necesario para llevarlas a cabo y especificar las acciones que el sistema debe realizar.

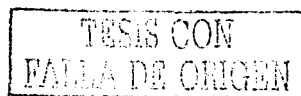
Se identifican los casos de uso examinando las necesidades de los usuarios para establecer la utilidad que debe tener el sistema.

Arquitectura

"La arquitectura es una vista de los modelos del sistema, de los modelos de casos de uso, análisis, diseño, implementación y despliegue. La descripción de la arquitectura describe partes del software que es importante que comprendan todos los desarrolladores." [Jacobson]

El establecimiento inicial de la arquitectura comprende una especificación cuidadosa para conseguir seguridad de que el proyecto progresará con éxito y terminará con satisfacción. El sistema se desarrolla basado en los casos de uso establecidos en la arquitectura, en el inicio se eligen los más importantes que son los que ayudarán mejor en el diseño de la arquitectura.

A medida que se va estableciendo la arquitectura se hace más estable, ya que desde el principio se contemplaron los casos de uso importantes. La descripción de la arquitectura debe mantenerse actualizada durante la vida del sistema para reflejar los cambios que son relevantes.



Iteraciones

Con un ciclo de vida iterativo los resultados son tangibles por que cada iteración aporta un incremento y demuestra la reducción de los riesgos con los que se relaciona, los productos de las iteraciones se presentan a los clientes y usuarios para validar la funcionalidad y hacer una correcta retroalimentación.

La iteración controlada, reduce desde el principio del ciclo de vida, los riesgos que pueden amenazar el progreso de desarrollo del proyecto. Las versiones internas tras las iteraciones posibilitan la retroalimentación de los usuarios y esto conlleva a su vez a la corrección temprana en el proyecto.

Las primeras iteraciones ayudan a comprender los riesgos, determinar la viabilidad, construir la arquitectura y realizar el análisis del negocio, las últimas iteraciones incorporan incrementos, pues se van añadiendo funcionalidades secundarias hasta llegar al producto final.

La estrategia de las iteraciones radica en que se van realizando las actividades del proceso de desarrollo planificando un poco, especificando, diseñando e implementando un poco e integrando, probando y ejecutando un poco en cada iteración. En cada iteración se va obteniendo retroalimentación que permite ajustar pequeñas desviaciones en las siguientes iteraciones.

Fases

El Proceso Unificado se divide en cuatro fases, las cuales son: inicio, elaboración, construcción y transición.

En la fase de inicio, el objetivo principal es establecer el análisis del negocio, el criterio esencial es conocer la viabilidad mediante la identificación de riesgos, la creación de una arquitectura a partir del desarrollo de un subconjunto importante de requerimientos, desde el modelado de los casos de uso, la estimación inicial de costo, esfuerzo, calendario y calidad del producto, el análisis del negocio se hace para determinar si el proyecto merece la pena.

En la fase de elaboración, el objetivo principal es obtener una arquitectura para guiar el sistema. El criterio esencial es conocer la capacidad para construir el sistema dentro de un marco económico mediante la identificación de riesgos, la especificación de la mayoría de los casos de uso que representan la funcionalidad, la arquitectura a seleccionar deberá concordar con las proporciones de una línea base y la preparación del plan del proyecto con suficiente detalle como para guiar la fase de construcción.

En la fase de construcción, el objetivo principal es hacer posible la operación del sistema. Se detallan los casos de uso, se integran los subsistemas y se hacen pruebas. Se desarrollan los componentes de acuerdo a la arquitectura.

En la fase de transición, el objetivo principal es hacer la entrega del sistema al usuario final. El sistema tiene la capacidad para ejecutarse en el ambiente de los usuarios, se realizan cambios menores revisando que se ha cumplido con los requerimientos a través de las pruebas de aceptación. Se llevan a cabo las actividades relativas a instalar y configurar el sistema para operar en el ambiente de producción y se hace la entrega de manuales y recursos adicionales.

Flujos de trabajo

Un flujo de trabajo se define como un conjunto de actividades que producen un resultado específico.

Los flujos que comprende el Proceso Unificado se mencionan enseguida.

Requerimientos

El flujo de trabajo de requerimientos tiene el propósito de indicar como será el desarrollo del sistema, mediante la descripción adecuada de las necesidades del cliente. Los pasos a considerar para obtener los requerimientos comienzan con la enumeración de necesidades, de mayor a menor prioridad, se debe comprender el contexto del sistema.

La planeación del trabajo se hace en base a los requerimientos y a partir de éstos se propone la estructura que tendrá el sistema, se estima el costo de implementación basado en recursos y tiempo, las prioridades y el nivel de riesgo asociado a la implementación de cada requerimiento.

La técnica para identificar requerimientos del sistema se basa en los casos de uso que capturan tanto requerimientos funcionales como no funcionales. Cada caso de uso representa una funcionalidad del sistema.

Análisis

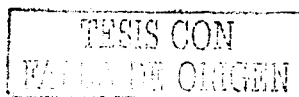
El flujo de trabajo de análisis tiene la finalidad de profundizar en los requerimientos y crear una estructura adecuada para conseguir que se comprendan las funciones del sistema. En este flujo es indispensable modelar los casos de uso. La estructura de los casos de uso es creada como una especificación de funcionalidad completa que proporcione una representación clara e intuitiva.

En este flujo se identifican los detalles de los casos de uso especificándolos con el lenguaje de los desarrolladores, se hace el detalle en relación al funcionamiento interno del sistema, se estructuran las clases y paquetes esbozando la funcionalidad del sistema.

Diseño

Los propósitos del flujo de trabajo de diseño son adquirir una comprensión profunda de los aspectos relacionados con los requerimientos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes, sistemas operativos, tecnologías de distribución y concurrencia, tecnologías de interfaz de usuario, tecnologías de gestión de transacciones, etc. En el diseño se crea un punto de partida para actividades de implementación. El diseño debe tener la propiedad de descomposición para que la implementación se pueda realizar en partes manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo.

“El diseño es el centro de atención al final de la fase de elaboración y el comienzo de las iteraciones de construcción. Esto contribuye a una arquitectura estable y sólida y a crear un plano del modelo de implementación.”[Jacobson]



Implementación

El flujo de trabajo de implementación tiene el propósito de crear el código en base a la arquitectura. El modelo de implementación describe como se organizan los componentes de acuerdo con los mecanismos de estructuración, modularización y dependencias entre componentes.

Pruebas

El flujo de trabajo de pruebas tiene como propósito hacer la planeación de las pruebas necesarias en cada iteración, pruebas de integración y pruebas de sistema. Las pruebas de integración son necesarias para cada construcción dentro de una iteración, mientras que las pruebas de sistema son necesarias sólo al final de la iteración.

Se diseñan e implementan las pruebas creando casos de prueba que especifican qué probar, con qué datos y cómo se deben hacer dichas pruebas.

Proceso Unificado de Rational

El Proceso Unificado de Rational es un proceso de ingeniería de software creado por la empresa Rational que provee un enfoque disciplinado para la asignación de tareas y responsabilidades en una empresa de software. Tiene el objetivo de asegurar la producción de software de calidad que cumpla con las necesidades de los usuarios finales.

El Proceso Unificado de Rational agrega los flujos de trabajo de modelado del negocio, distribución, administración de configuración y cambios, administración del proyecto y el de ambiente.

Los flujos que agrega el Proceso Unificado de Rational se mencionan enseguida.

Modelado del negocio

Tiene el propósito de establecer el conocimiento de la estructura y la dinámica de la organización. Entender los problemas e identificar mejoras potenciales. Asegurar que los clientes, usuarios y desarrolladores tienen un conocimiento común de la organización. Obtener los requerimientos del sistema necesarios para apoyar las actividades de desarrollo.

Distribución

Tiene el propósito de describir las actividades relacionadas con asegurar que el software estará disponible para los usuarios finales. Se da énfasis a las pruebas del producto en el sitio de desarrollo antes de que se haga la liberación para hacer la entrega a los usuarios finales.

Administración de configuración y cambios

Tiene el propósito de monitorear y mantener la integridad de los recursos del proyecto. Durante el ciclo de vida de desarrollo se crean muchos artefactos. El desarrollo de

estos artefactos es una labor importante y estos representan una inversión, por ello, estos deben ser guardados y estar disponibles para reutilizarlos.

Al mismo tiempo el equipo del proyecto debe monitorear la evolución del producto para manejar solicitudes de cambio no importando de donde procedan y después implementarlas de una manera consistente en los artefactos.

Administración del proyecto

Tiene el propósito de proporcionar las guías para llevar a cabo la administración del desarrollo del software. Este flujo esta enfocado a equilibrar los objetivos, administrar los riesgos y superar las restricciones para entregar un producto que satisfaga las necesidades de clientes y usuarios.

Ambiente

Tiene el propósito de ofrecer soporte al desarrollo de software con la configuración del proceso y sus herramientas, tiene actividades para evaluar a la organización de desarrollo de software, configurar y mejorar el proceso, seleccionar herramientas y ofrecer servicios técnicos para apoyar el proceso.

Diagramas de UML

El Lenguaje de Modelado Unificado (UML) es un lenguaje que incluye un repertorio de conceptos, términos y símbolos para construir modelos visuales en los cuales se constituye el conocimiento para desarrollar aplicaciones informáticas. Permite especificar la funcionalidad de un sistema a través de diferentes vistas.

UML es un lenguaje público surgido por un esfuerzo de personas y empresas expertas en el campo de la ingeniería de software. Pretende abordar los problemas actuales del desarrollo de software, tales como, el tamaño, distribución, concurrencia y desarrollo en equipo.

Tiene el propósito de ser simple y expresivo con la capacidad de modelar una gran gama de sistemas.

Las vistas de un modelo muestran diferentes perspectivas de un sistema para que sea comprendido ampliamente. Los símbolos que se utilizan tienen un significado semántico a través del cual se entienda el contexto en el cual están ubicados los elementos del modelo.

Este lenguaje combina conceptos comúnmente aceptados por muchos métodos orientados a objetos y constituye definiciones para conceptos usando símbolos y terminología. Los conceptos y notación se utilizan en las diferentes etapas de desarrollo de sistemas. Tiene la capacidad de que puede ser ampliado hacia nuevas situaciones.

Una ventaja es que aún siendo muy amplio debido a que tiene varios tipos de modelos, se puede ir aprendiendo de acuerdo a las necesidades.

Diagrama de clases

Se utilizan para modelar la vista de diseño estática en un sistema, se modela el vocabulario del sistema, se modelan las colaboraciones y los esquemas.

Los diagramas de clases son los más utilizados en el modelado de sistemas orientados a objetos. Un diagrama de clases muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones.

Diagrama de objetos

Se utilizan para modelar la vista estática instantánea del sistema en un momento concreto y la representación de un conjunto de objetos, su estado y sus relaciones.

Los diagramas de objetos modelan las instancias de los elementos contenidos en los diagramas de clases.

Diagrama de casos de uso

Se utilizan para modelar el comportamiento de un sistema, subsistema o clases en donde muestran un conjunto de casos de uso, actores y sus relaciones. La mayoría de las veces, esto implica modelar el contexto del sistema, subsistema, clase o el modelado de los requerimientos.

Los diagramas de casos de uso son importantes para visualizar, especificar y documentar el comportamiento de un elemento. Estos diagramas facilitan que los sistemas, subsistemas y clases sean comprensibles, al presentar una vista externa de cómo pueden utilizarse estos elementos en un contexto dado.

Diagrama de interacciones

Muestra la interacción de un conjunto de objetos incluyendo los mensajes que se envían entre ellos. Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema. La mayoría de las veces esto implica modelar instancias concretas, interfaces o componentes.

Los diagramas de interacción se utilizan para visualizar, especificar, construir y documentar la dinámica de una sociedad particular de objetos. También se puede utilizar para modelar un flujo de control particular de un caso de uso.

Diagrama de actividades

Se utilizan para modelar los aspectos dinámicos de un sistema. La mayoría de las veces, esto implica modelar los pasos secuenciales y posiblemente concurrentes de un proceso. Con un diagrama de actividades también se puede modelar el flujo de un objeto conforme cambia de estado en diferentes puntos del flujo de control.

Los diagramas de actividades pueden utilizarse para visualizar, especificar, construir y documentar la dinámica de una sociedad de objetos o para modelar el flujo de control de una operación.

Diagrama de estados

Se utilizan para modelar aspectos dinámicos de un sistema. La mayoría de las veces, esto supone el modelado del comportamiento de objetos reactivos. Un objeto reactivo es aquél para el que la mejor forma de caracterizar su comportamiento es señalar cuál es su

respuesta a los eventos lanzados desde fuera de su contexto. Un objeto reactivo tiene un ciclo de vida bien definido, cuyo comportamiento se ve afectado por su pasado. Los diagramas de estados pueden asociarse a las clases, los casos de uso o a sistemas completos para visualizar, especificar, construir y documentar la dinámica de un objeto individual.

Diagrama de componentes

Se utilizan para modelar los aspectos físicos de los sistemas orientados a objetos, muestra la organización y las dependencias entre un conjunto de componentes. Los diagramas de componentes se utilizan para modelar la vista de implementación estática de un sistema. Esto implica modelar las cosas físicas que residen en un nodo, tales como ejecutables, bibliotecas, tablas, archivos y documentos.

Diagramas de despliegue

Se utilizan para modelar los aspectos físicos de los sistemas orientados a objetos, un diagrama de despliegue muestra la configuración de nodos que participan en la ejecución de componentes que residen en ellos.

Los diagramas de despliegue se utilizan para modelar la vista de despliegue estática de un sistema. La mayoría de las veces, esto implica modelar la topología del hardware sobre el que se ejecuta el sistema. Los diagramas de despliegue son fundamentalmente diagramas de clases que se ocupan de modelar los nodos de un sistema.

Instituto de Investigaciones Bibliográficas

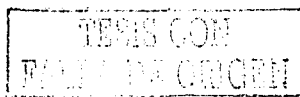
El Instituto de Investigaciones Bibliográficas tiene a su cargo la Biblioteca y la Hemeroteca Nacionales de México, responsabilidad que se otorga a la UNAM en 1929. Cuenta con las características propias de un centro de investigaciones, el Instituto mismo y con las particularidades de una dependencia de servicios debido a la Biblioteca y la Hemeroteca Nacionales.

Departamento de Informática

El Instituto cuenta con un Departamento de Informática, el cual, realiza diversas funciones como, administración de redes y equipo de cómputo, instalación de paquetes comerciales, soporte técnico, capacitación de personal y desarrollo de sistemas.

El área de desarrollo de sistemas es la que ha demandado mayor atención a través de los años, inicialmente los sistemas realizados eran simples, cumplían holgadamente las expectativas que los usuarios tenían, ya que en muchos casos eran usuarios que solicitaban un sistema por primera vez, con algunas cuantas funcionalidades.

La complejidad de los sistemas se ha incrementado un poco debido a que ahora los usuarios quieren que los sistemas de reciente creación se vinculen con los existentes, desean agregar datos multimedia, que se puedan acceder a través de Internet, que se ejecuten directamente de discos compactos sin instalación previa, que aseguren la privacidad de los datos, entre otras.



Otros aspectos importantes que han ocurrido en el desarrollo de sistemas son, el uso de diferentes plataformas, uso de diferentes versiones de lenguajes de programación y varios lenguajes, lo anterior complica la situación para hacer vinculaciones entre diferentes sistemas, lo cual, crea repetición de actividades y redundancia de datos que posteriormente se convierten en cuestiones difíciles de corregir que contaminan el desarrollo de nuevos sistemas.

La funcionalidad de los sistemas tiende a cambiar debido a que los usuarios piden soluciones a problemas pensando en que no habrá más requerimientos, la realidad es que siempre requieren más, surgen nuevas situaciones de trabajo, otras veces es necesario cambiar labores sencillas por otras más complejas, incluso la funcionalidad del nuevo sistema crea en el usuario un impulso para pedir más cosas, es decir, se crean en el usuario nuevos intereses que antes no eran importantes o estaban escondidos.

Capítulo 2

Planteamiento de la adaptación

El propósito de esta tesis es hacer la adaptación de los flujos de trabajo de ambiente, distribución y pruebas del Proceso Unificado al Departamento de informática del Instituto de Investigaciones Bibliográficas de la UNAM.

El alcance de la adaptación de estos tres flujos es colaborar en la integración de los demás flujos adaptados del Proceso Unificado que se encuentran en [Sánchez] con requerimientos, análisis y diseño, y en [Ramírez] con implementación, administración del proyecto, administración de cambios y configuración. Las actividades que se señalan en los nueve flujos se distribuyen en las cuatro fases del Proceso Unificado para obtener el proceso completo de desarrollo de sistemas del departamento de informática del Instituto de Investigaciones Bibliográficas de la UNAM.

La razón de que únicamente sean tres flujos de trabajo los que se adaptan en este trabajo, es porque se trata de múltiples actividades, artefactos y roles los que se deben estudiar para obtener el proceso completo, labor que para una sola persona sería compleja y difícil pensando en que deseara abarcar todos los flujos.

Para acoplar las actividades de los nueve flujos de trabajo se realizaron múltiples reuniones en donde se establecieron lineamientos para poder obtener en tres trabajos de tesis una solución completa con la ventaja de poder trabajar de manera independiente en cada tesis con los flujos de trabajo asignados.

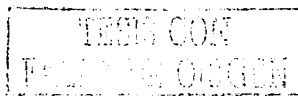
Cada persona fue realizando el análisis de cada flujo de trabajo, estudiando las actividades planteadas por el Proceso Unificado, comparándolas con las actividades que se hacen actualmente en el departamento para plantear la adaptación.

Se establecieron estándares para crear los diagramas y artefactos. Se utilizaron diagramas de actividades para modelar las actividades del proceso y se diseñaron los artefactos de acuerdo a las necesidades del departamento y al formato indicado.

Después de que se terminaron de adaptar todos los flujos de trabajo se ordenaron las actividades de cada flujo en las cuatro fases del Proceso Unificado, posteriormente se introdujeron los modelos de todos los flujos de trabajo en una herramienta que muestra a través de diagramas de actividades el flujo de inicio a fin del proceso de desarrollo de sistemas.

La intención de crear éste proceso es utilizarlo en la práctica, para que sirva como guía en el desarrollo de sistemas. Debe también, ayudar a realizar las actividades de una manera organizada, favorecer la comunicación entre los miembros del departamento que realizan las actividades de desarrollo, servir para conocer en algún momento en que punto del proceso se encuentra el sistema que se está construyendo y, finalmente, que ayude a hacer más fácil la integración de nuevo personal disminuyendo la curva de aprendizaje sobre la forma de trabajo y tipo de sistemas que se hacen en el departamento.

Los tres flujos que se adaptan en este trabajo son tomados del Proceso Unificado de Rational, sin embargo, siempre se tomará el nombre de Proceso Unificado para hacer la referencia, aunque se está haciendo hincapié que sólo se tomó en cuenta la especificación de los flujos de Rational.



Objetivos

- Adaptar los flujos de trabajo ambiente, distribución y pruebas del Proceso Unificado al Departamento de Informática del Instituto de Investigaciones Bibliográficas.
- Analizar al Departamento de Informática para conocer de que manera se realizan las actividades de desarrollo de sistemas.
- Adecuar las actividades y pasos de esos flujos al departamento de informática.
- Adecuar los roles de esos flujos al Departamento de Informática.
- Diseñar y crear los artefactos de acuerdo a las necesidades del Departamento de Informática.
- Integrar los flujos desarrollados con los demás flujos adaptados de [Sánchez] y [Ramírez].
- Colaborar en la creación de una herramienta que contenga la descripción de todo el proceso para que sirva como guía en el desarrollo de sistemas del Departamento de Informática.

Análisis para hacer la adaptación

Para hacer la adaptación de los flujos de trabajo al Departamento de Informática es necesario ubicarlo en un contexto. El departamento tiene a su cargo además del área de desarrollo de sistemas otras más, como el soporte técnico, la administración y seguridad de redes, administración de servidores, capacitación de personal, instalación de programas y equipo de cómputo, asesorías, etc., y el personal del departamento debe afrontar las responsabilidades de todas estas áreas. El departamento cuenta con pocas personas para hacer las actividades de desarrollo.

Para establecer el proceso de desarrollo de sistemas fue indispensable analizar al departamento para conocer las habilidades y actitudes de las personas, los recursos con los que cuentan, el tipo de sistemas que se desarrollan y la forma en que realizan las actividades. Este tipo de análisis es una actividad importante del flujo de trabajo ambiente, esta actividad se llama **“Evaluar la organización de desarrollo actual”** y se puede leer con detalle en el apéndice A.

El propósito del flujo del trabajo ambiente es realizar las actividades para configurar el proceso, crear los artefactos acordes a la empresa de desarrollo de software y proporcionar soporte al equipo de desarrollo.

Valoración del departamento de informática

El propósito de esta actividad es describir el estado actual de la empresa de software en términos de su proceso actual, herramientas disponibles, competencias y actitudes de las personas, clientes, competidores, tendencias técnicas, problemas y áreas de mejora.

La valoración del departamento se realizó como lo establece el flujo de trabajo ambiente.

Se obtuvieron los siguientes aspectos:

- El personal que hace los sistemas no cuenta con una forma establecida para realizar el trabajo. Cada sistema lo desarrollan de acuerdo a las experiencias acumuladas y a las habilidades aprendidas por desarrollos anteriores.
- Las personas que actualmente realizan labores de desarrollo de sistemas en el departamento están convencidas de que crear un proceso de acuerdo a sus necesidades les facilitará la manera de hacer su trabajo tanto para los nuevos proyectos como para el mantenimiento.
- Las personas tienen habilidades para llevar a cabo cualquier fase del ciclo de desarrollo de sistemas, por lo que, están motivadas a utilizar el proceso.
- Las herramientas disponibles para realizar el desarrollo de sistemas son principalmente algunos lenguajes de cuarta generación, algunos manejadores de bases de datos, herramientas que tienen que ver con sitios de Internet, diferentes sistemas operativos y algunos otros elementos de menor importancia. En otras palabras, hay variedad en plataformas y para el almacenamiento de datos.
- No cuentan con herramientas que les apoyen en actividades de administración de proyectos, ni mucho menos cuentan con herramientas de apoyo de gestión de procesos.
- El perfil de los usuarios de los sistemas es heterogéneo, hay usuarios novatos y especializados, lo cual encierra un aspecto muy especial de algunos sistemas.
- Los roles actuales de las personas que laboran en el departamento son de dos tipos: una sola persona se encarga de todo el sistema desde el inicio hasta el final incluyendo el mantenimiento; y, conformación de pequeños equipos en donde se trabaja repartiendo el trabajo para hacer algunas partes del sistema y después juntarlas.
- En este departamento todas las personas son importantes, primero porque conocen el desarrollo de sistemas y segundo porque son un número reducido.
- Las habilidades que tienen las personas de desarrollo de sistemas pueden ser estimadas entre la clasificación de personal calificado hasta el nivel de experto, sin descartar que pueden llegar en algún momento personas de apoyo que caigan en la categoría de aprendiz o de novato.

TESIS CON FALLA DE ORIGEN

- La capacidad para realizar el cambio en el Departamento de Informática parece ser prometedora, esto se expone así debido a que las personas que laboran allí ya han tenido un acrcamiento a la información de procesos.
- El departamento no cuenta con ningún tipo de descripción de procesos, ni de guías y ni de plantillas de documentos.
- Los sistemas que se hacen son principalmente bibliográficos, hemerográficos y biográficos. Además existe el interés que estos sistemas vayan expandiéndose a través de las diferentes plataformas para incluir a más usuarios. Otro hecho que es relevante conocer es que los sistemas tienden también a querer integrarse o vincularse con operaciones de otros sistemas.
- Algunos de los problemas técnicos que se dan en el departamento es el manejo de versiones del software utilizado, para crear bases de datos, archivos, bibliotecas de objetos, funciones programadas y más. Lo que quiere decir, que al haber diferentes versiones de elementos para el desarrollo de sistemas se llegan a provocar mezclas raras en una computadora que hace que diferentes sistemas no funcionen como se espera.
- No cuentan con herramientas comerciales como el Proceso Unificado de Rational que apoyen el proceso, mencionan que éstas son muy caras para poder adquirirlas.
- Los proyectos pocas veces se terminan a tiempo, esto se atribuye a que las personas del departamento no mantienen su atención completamente en el desarrollo de sistemas, también les preocupan otras áreas y por ello dejan de hacer las actividades concernientes a los sistemas por varios días.
- No hay un presupuesto definido exclusivamente para cada proyecto, pero se expresó por parte del personal, que se invierten muchas horas en el mantenimiento y mencionan que eso implica que los sistemas salen más caros.
- La raíz de los problemas se debe a que el número de peticiones de sistemas y requerimientos del sistema está sobrepasando el número de horas de desarrollo disponibles por las personas del departamento.

Estos aspectos son consideraciones generales para hacer la adaptación de los tres flujos de trabajo que serán tratados en los tres capítulos siguientes. En cada flujo se destacan las consideraciones particulares para hacer el ajuste de artefactos, roles y actividades que propone el Proceso Unificado de Rational; el reto es crear artefactos que satisfagan las necesidades del personal que realiza las actividades de desarrollo de sistemas del Departamento de Informática.

Capítulo 3

Flujo de trabajo ambiente

El propósito de este flujo es realizar las actividades para configurar el proceso, crear los artefactos acordes a la empresa seleccionada y proporcionar soporte al equipo de desarrollo. La descripción a detalle de acuerdo al Proceso Unificado de Rational se encuentra en el Apéndice A.

En este capítulo se explica como es el flujo de trabajo ambiente, las consideraciones para hacer la adaptación y el resultado de la adaptación de cada actividad y sus pasos. Se hace una explicación de los artefactos y para los roles se hace una descripción y se describe el perfil.

Para explicar cada actividad se tiene el siguiente formato: se pone del lado izquierdo la etiqueta RUP, el nombre de la actividad y el nombre del rol responsable de la actividad, de acuerdo a como lo establece RUP. Después se pone la etiqueta ADAPTACION y el nombre de la actividad y enseguida las consideraciones para hacer la adaptación. Después se utiliza la etiqueta PDSIIB para designar al proceso de desarrollo de sistemas del Instituto de Investigaciones Bibliográficas en el Departamento de Informática, y se pone el nombre de la actividad como quedó en el proceso resultante y el nombre del rol responsable.

Cuando finaliza la descripción de la actividad se pone a nivel de pasos cada actividad con la etiqueta PDSIIB y el nombre del paso con su descripción.

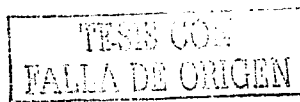
Actividades

RUP <Actividad> Preparar el ambiente para el proyecto
<Rol> *Ingeniero de procesos, especialista en herramientas*

Los propósitos de esta actividad son evaluar a la organización de desarrollo de software, perfilar el desarrollo del proceso, listar herramientas candidato para los desarrollos y crear las plantillas para los artefactos.

Para realizar esta actividad se hace lo siguiente:

- Valorar a la empresa.
- Identificar a las personas afectadas por el desarrollo de sistemas.
- Describir a la empresa interiormente.
- Identificar a las personas importantes para la creación del proceso.
- Identificar las razones para realizar cambios en la manera en que actualmente se desarrollan los sistemas.
- Identificar las habilidades y actitudes de las personas.
- Hacer la estimación para realizar los cambios en la empresa debido al nuevo proceso.
- Analizar la descripción del proceso de desarrollo actual de la empresa.
- Caracterizar los tipos de proyectos y aplicaciones típicas que desarrolla la empresa y los proyectos que se encuentran en producción.



- Identificar las herramientas de apoyo que usan actualmente en los proyectos y las que hacen falta.
- Identificar problemas en la empresa al realizar los proyectos.
- Escribir las conclusiones después de analizar la información.
- Hacer recomendaciones acerca de cómo debe llevarse a cabo el nuevo proceso y el uso de herramientas.

Cuando esta actividad se ha completado se conoce: el estado actual de la empresa, los tipos de personas involucradas y sus niveles de competencia, habilidades y motivación, las herramientas de apoyo que actualmente utiliza la empresa y el software que actualmente utiliza para el diseño y descripción del proceso.

Al hacer la evaluación del estado actual se tiene información suficiente para: utilizar el estado actual como entrada para configurar el proceso, identificar además las áreas que necesitan mejoras primero, explicar a los patrocinadores porque necesitan cambiar el proceso y crear motivación en las personas de la empresa para considerar a quien se afecta directa e indirectamente por los cambios.

ADAPTACIÓN <Actividad> Preparar el ambiente para el proyecto

Esta actividad fue muy importante para hacer la adaptación, ya que en ella se describe como hay que hacer el estudio de la organización para comprender como se realizan las actividades de desarrollo de sistemas; después se crea el proceso de desarrollo de software tomando los parámetros del estudio y las necesidades de la organización.

Esta actividad cambió de nombre y de propósito debido a que se realizó sólo para analizar al Departamento de Informática y en base a la descripción obtenida de la evaluación del departamento, se ha creado el nuevo proceso y ya no será necesaria, el nuevo nombre es "Conocer el ambiente para el proyecto", la cual, describe los estándares que deben utilizarse para el proceso de desarrollo de sistemas del Departamento de Informática.

PDSIIB <Actividad> Conocer el ambiente para el proyecto

<Rol> Todos

Conocer el ambiente para el proyecto consiste en leer los estándares del proceso de desarrollo para relacionarse con la manera en que se deben hacer las actividades del proyecto.

Tiene los siguientes artefactos:

- Estándar del proceso de desarrollo de sistemas del IIB (PDSIIB) [Salazar]
- Estándar para modificar el proceso [Salazar]
- Estándar de documentación [Sánchez]
- Estándar de herramientas [Salazar]
- Lista de actividades, artefactos y roles [Ramírez]
- Glosario [Sánchez]
- Estándar de administración de riesgos [Ramírez]

- Estándar de evaluación [Ramírez]
- Estándar de solución a problemas [Ramírez]
- Estándar de aceptación [Ramírez]
- Estándar de administración de cambios [Ramírez]
- Estándar de administración de configuración [Ramírez]
- Estándar de requerimientos [Sánchez]
- Estándar de análisis y diseño [Sánchez]
- Estándar de base de datos [Sánchez]
- Estándar de codificación [Ramírez]
- Estándar de pruebas [Salazar]
- Estándar de distribución [Salazar]

RUP <Actividad> Preparar el ambiente para una iteración

<Rol> *Ingeniero de procesos, especialista en herramientas*

El propósito de esta actividad es describir el proceso de desarrollo de software para un proyecto o proyectos de la organización. Esta actividad es igual a la que se describió en preparar el ambiente para el proyecto, con la diferencia de que se debe de hacer para cada nuevo proyecto.

ADAPTACIÓN <Actividad> Preparar el ambiente para una iteración

Esta actividad cambió de nombre debido a que se realizó sólo para conocer las herramientas con las que cuenta el Departamento de Informática para desarrollar los sistemas, el nuevo nombre es "Preparar el ambiente para un ciclo", la cual, describe los pasos para indicar las herramientas a comprar y la configuración de herramientas para cada ciclo de desarrollo del proyecto.

RUP <Actividad> Preparar las guías del proyecto para una iteración

<Rol> *Analista de procesos, analista de sistemas, diseñador de pruebas, arquitecto de software, especialista en herramientas, escritor técnico*

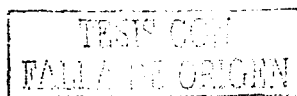
Las guías serán modificadas si existen variaciones y necesidades específicas en los sistemas a desarrollar.

ADAPTACIÓN <Actividad> Preparar las guías del proyecto para una iteración

Esta actividad se realizó una sola vez para cada flujo de trabajo, sin embargo en esta adaptación el nombre de guía no existe, en su lugar se utiliza la palabra estándar.

Esta actividad es manejada con el artefacto "Estándar para modificar el proceso" donde se especifica la manera en que se debe proceder para cambiar un artefacto, actividad, paso o rol del proceso. Para ello, se utiliza una "Solicitud de cambio al PDSIIB".

Para modificar algún elemento del proceso se debe establecer un comité que tiene la responsabilidad de analizar, evaluar y modificar el proceso.



RUP <Actividad> Apoyar el ambiente durante una iteración

<Rol> Administrador del sistema

Esta actividad tiene como propósito apoyar en el proyecto de desarrollo con el hardware y software necesario, realizar respaldos, ofrecer soporte de redes, crear y reproducir documentos y otros más.

ADAPTACIÓN <Actividad> Apoyar el ambiente durante una iteración

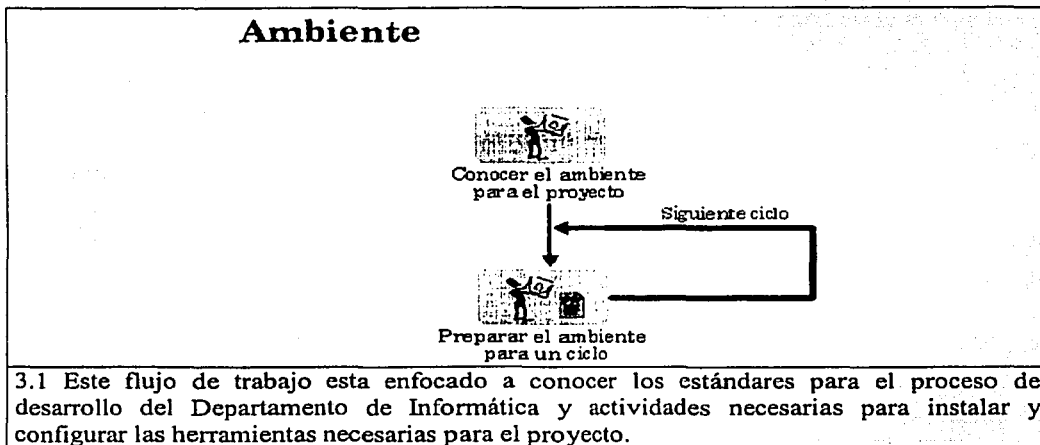
Esta actividad es indispensable para cualquier proyecto de desarrollo y siempre se hace, sin embargo, para la adaptación no se considera este rol debido a que no hay una persona que se dedique específicamente a apoyar el proyecto. Las cuestiones marcadas por ésta actividad las desempeña el mismo responsable del sistema.

PDSIIB <Actividad> Preparar ambiente para un ciclo

<Rol> Especialista en herramientas

Preparar el ambiente para un ciclo consiste en hacer una solicitud de compra de herramientas al Departamento de Informática y configurar las herramientas que se van a utilizar en el desarrollo de los casos de uso especificados en el ciclo.

El nuevo flujo de trabajo ambiente se puede apreciar en la figura 3.1.



PDSIIB <Actividad> Conocer el ambiente para el proyecto
<Rol> Cualquier rol

PDSIIB <Paso> Leer estándares de ambiente

Consiste en examinar el “Estándar del proceso de desarrollo de sistemas del IIB”, el “Estándar para modificar el proceso”, el “Estándar de documentación”, el “Estándar de herramientas”, la “Lista de actividades, herramientas y roles” y el “Glosario”.

PDSIIB <Paso> Leer estándares de administración del proyecto

Consiste en examinar el “Estándar de administración de riesgos”, el “Estándar de evaluación”, el “Estándar de solución de problemas” y el “Estándar de aceptación”.

PDSIIB <Paso> Leer estándares de administración de configuración

Consiste en examinar el “Estándar de administración de cambios” y el “Estándar de administración de configuración”.

PDSIIB <Paso> Leer estándares de requerimientos

Consiste en examinar el “Estándar de requerimientos”.

PDSIIB <Paso> Leer estándares de análisis y diseño

Consiste en examinar el “Estándar de análisis y diseño” y el “Estándar de base de datos”.

PDSIIB <Paso> Leer estándares de implementación

Consiste en examinar el “Estándar de implementación” que establece la forma en que se debe escribir el código de los lenguajes de programación establecidos para este proceso.

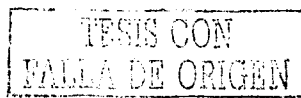
PDSIIB <Paso> Leer estándares de pruebas

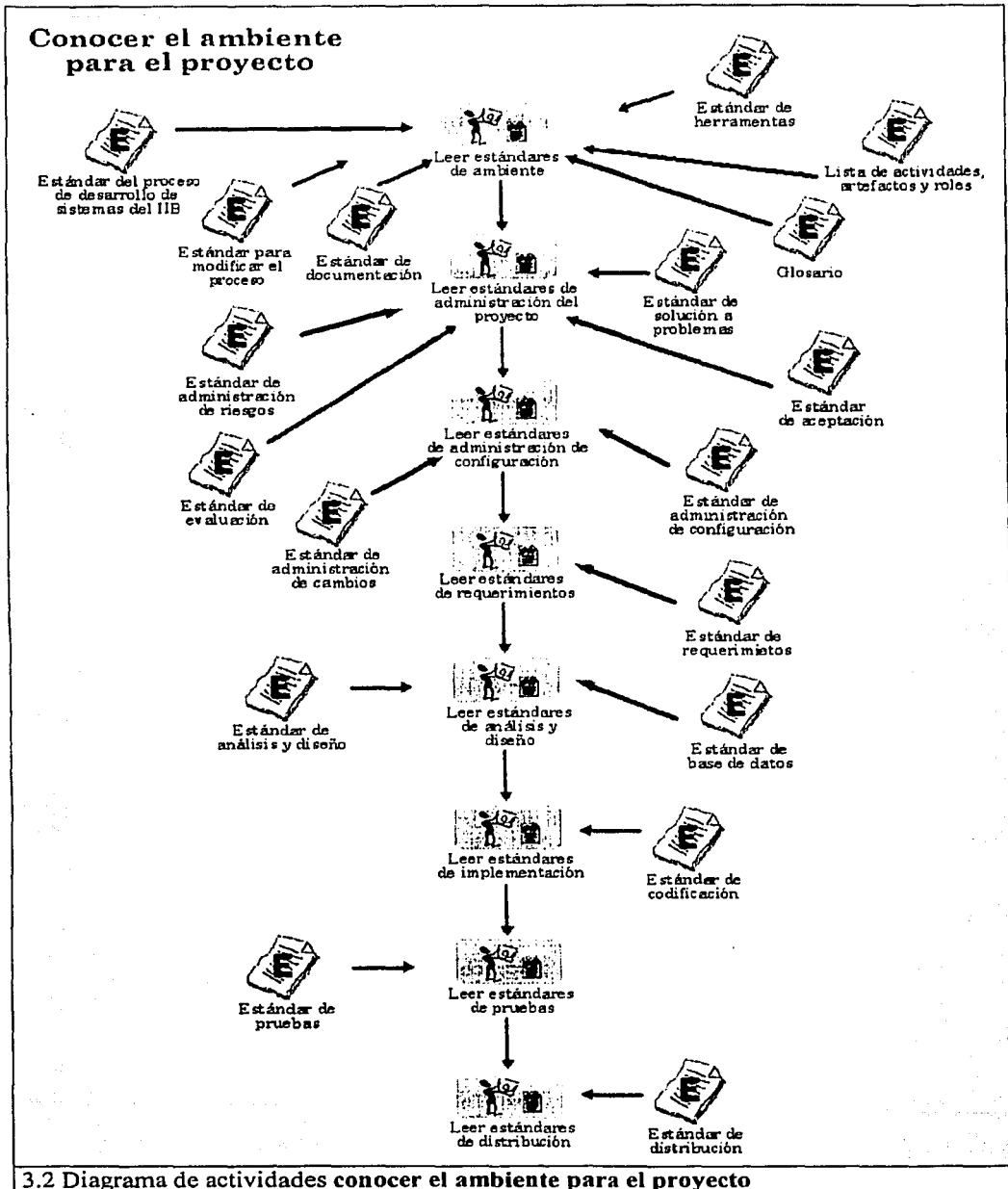
Consiste en examinar el “Estándar de pruebas”.

PDSIIB <Paso> Leer estándares de distribución

Consiste en examinar el “Estándar de distribución”.

La actividad **Conocer el ambiente para el proyecto** se puede apreciar en el diagrama 3.2.





3.2 Diagrama de actividades conocer el ambiente para el proyecto

PDSIIB <Actividad> Preparar el ambiente para un ciclo
<Rol> Especialista en herramientas

PDSIIB <Paso> Reunir artefactos de entrada

Reunir los artefactos de entrada necesarios correspondientes a esta actividad. Los artefactos son: Visión inicial y diseño de la arquitectura.

Del artefacto "Visión inicial" se identifican las herramientas que se deben instalar antes de que comience la actividad "Iniciar ciclo".

Del artefacto "Diseño de la arquitectura, sección 3" se identifican las herramientas que se deben instalar antes de comenzar la actividad "Establecer modelo de implementación".

PDSIIB <Paso> Identificar las herramientas a comprar

Consiste en especificar en el artefacto "Herramientas a comprar" los nombres de las herramientas que deberán ser adquiridas indicando la versión y los nombres de los sistemas operativos donde deben operar. Este artefacto lo firma el especialista en herramientas y lo entrega al Jefe del Departamento de Informática.

Las herramientas a comprar se toman del artefacto "Visión inicial".

Cuando las herramientas han sido compradas se debe registrar el nombre y versión en el artefacto "Estándar de herramientas".

PDSIIB <Paso> Indicar la configuración de las herramientas

Consiste en registrar la configuración de las herramientas en el artefacto "Configuración de herramientas", se escribe el nombre de la herramienta y las acciones para hacer la configuración.

En el artefacto "Estándar de herramientas" se encuentra a detalle en la sección 4 los "Procedimientos de instalación de herramientas" y en la sección 5 los "Procedimientos de configuración de herramientas". Cuando la herramienta es nueva y se va a instalar y configurar por primera vez se debe anexar la información de instalación y configuración en las secciones 4 y 5 al "Estándar de herramientas".

PDSIIB <Paso> Instalar y configurar las herramientas

Consiste en hacer la instalación y configuración de las herramientas en los equipos de cómputo para el desarrollo del sistema en base a lo especificado en el artefacto "Configuración de herramientas".

Los procedimientos de instalación y configuración de herramientas se encuentran en el "Estándar de herramientas".

TESIS CON
FALLA DE ORIGEN

PDSIIB <Paso> Verificar la instalación de las herramientas

Consiste en revisar que la instalación y configuración de las herramientas son correctas, para ello, se utiliza el artefacto "Configuración de herramientas", si la configuración es correcta se pone la etiqueta "si" en la casilla de verificado, de lo contrario se pone la etiqueta "no".

El especialista en herramientas hace la verificación junto con el administrador del proyecto. Cuando todas las herramientas se han verificado el administrador del proyecto debe firmar el artefacto "Configuración de herramientas".

PDSIIB <Paso> Informar sobre las herramientas instaladas

Consiste en informar a todos los involucrados que ya pueden utilizar las herramientas para el desarrollo del sistema.

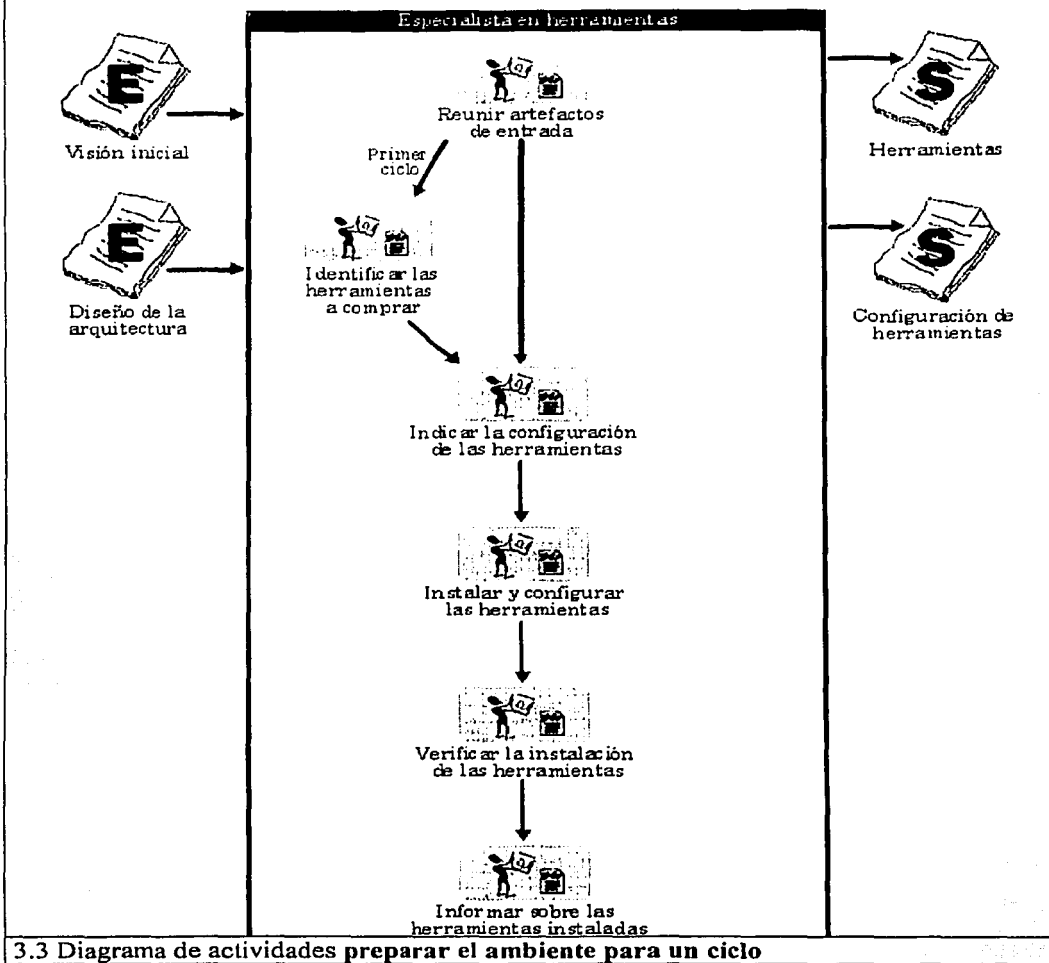
Informar verbalmente o por correo electrónico.

Artefactos de salida

Herramientas y configuración de herramientas.

La actividad **preparar el ambiente para un ciclo** se puede apreciar en el diagrama 3.3.

Preparar el ambiente para un ciclo



TRABAJA CON
VALLA DE ORIGEN

Roles

PDSIIB <Rol> Especialista en herramientas

Es el encargado de instalar las herramientas que se utilizaran en el proyecto. Esto es, sistema operativo, lenguajes de programación, manejadores de bases de datos, bibliotecas de objetos, procesadores de palabras, reporteadores, etc.

Perfil del rol

Tener conocimiento de ambientes de desarrollo de sistemas y la habilidad para instalar y configurar herramientas.

Artefactos

PDSIIB <Artefacto> Estándar del proceso de desarrollo de sistemas del IIB (PDSIIB)

En este artefacto se describe como está estructurado el proceso de desarrollo de sistemas del Departamento de Informática. Describe las fases que tiene el proceso, los flujos de trabajo, los diagramas de actividades, los niveles de los diagramas e indica cómo se utiliza este proceso.

La finalidad de tener este estándar es para que el personal conozca como utilizar el proceso para guiar las actividades de desarrollo de sistemas.

Este artefacto se muestra en la siguiente página.

TESIS CON
FALLA DE ORIGEN

Estándar del proceso de desarrollo de sistemas del IIB (PDSIIB)

El propósito de este artefacto es describir como esta estructurado el proceso de desarrollo de sistemas del Departamento de Informática del Instituto de Investigaciones Bibliográficas.

Este proceso de desarrollo de sistemas es el resultado de la adaptación de los flujos de trabajo del Proceso Unificado a las necesidades del Departamento de Informática.

Fases

El proceso tiene cuatro fases: inicio, elaboración, construcción y transición.

- **Inicio:** Es el conjunto de actividades para definir el alcance del sistema y determinar las funciones principales de acuerdo a los requerimientos.
- **Elaboración:** Es el conjunto de actividades para establecer en detalle la arquitectura del sistema e instalar las herramientas para el desarrollo del sistema.
- **Construcción:** Son el conjunto de actividades para construir el producto y hacer las pruebas de aceptación de acuerdo a los requerimientos.
- **Transición:** Son el conjunto de actividades para asegurar que el sistema estará disponible para la entrega a los usuarios finales, hacer pruebas y ajustes menores. Esta fase contempla el comienzo de otro nuevo ciclo del mismo proyecto cuando el sistema todavía no está terminado.

Flujos

El proceso tiene nueve flujos y cada actividad de los flujos se encuentran integrados en un diagrama de fases, es decir, las diferentes actividades del flujo se encuentran incluidas en alguna fase del proceso. Sin embargo existe la referencia del flujo al que corresponde una actividad.

Diagramas de actividades

Estos diagramas representan el flujo de actividades que se deben realizar durante el proceso de desarrollo de un sistema.

Niveles de diagramas:

- **Diagramas de fase:** Muestran las actividades que tiene una fase.
- **Diagramas por actividad:** Muestran los pasos que tiene una actividad.
 - En la parte izquierda del diagrama se muestran los artefactos de entrada que serán utilizados al realizar algún paso.
 - En la parte central del diagrama se muestran los pasos a realizar. Los pasos están delimitados en el diagrama por carriles que indican el rol responsable de llevar a cabo los pasos.
 - En la parte derecha del diagrama se muestran los artefactos de salida que han sido utilizados en algún paso.
 - Cuando en un artefacto de salida aparecen tres asteriscos ***, significa que el artefacto ya debe tener datos y que se van a agregar otros más.

Uso del PDSIIB

Para navegar a través del proceso se debe de pulsar sobre el nombre de una fase, actividad, paso o artefacto para obtener una descripción a mayor detalle.

Pulsar sobre el nombre de una fase significa obtener un nuevo diagrama que corresponde a las actividades de una fase.

Pulsar sobre el nombre de un artefacto significa obtener el documento real o una explicación de cómo se utiliza el artefacto.

Pulsar sobre el nombre de una actividad significa obtener el diagrama de actividad mostrando los pasos que corresponden a esa actividad.

Pulsar sobre el nombre de un paso significa obtener la explicación de lo que se debe hacer en ese paso.

Las flechas representan el flujo secuencial entre las actividades y entre los pasos.

Una línea horizontal en la que se juntan más de una flecha representa que las actividades o los pasos anteriores se pueden realizar de manera independiente.



PDSIIB <Artefacto> Estándar para modificar el proceso

En este artefacto se describe la manera en que debe ser modificado el proceso, aquí se indica que debe haber un comité responsable para hacer los cambios en el proceso.

Este estándar incluye el artefacto "Comité responsable del PDSIIB", donde se encuentran los nombres de las personas responsables de analizar, evaluar y modificar el proceso.

También incluye el artefacto "Solicitud de cambio al PDSIIB", que se utiliza para registrar el artefacto, actividad o paso que alguien desea se modifique o agregue en el proceso, se escribe la explicación, por la cual, se desea hacer la modificación y la fecha de la petición del cambio.

La finalidad de tener este estándar es para que exista un instrumento para que el proceso pueda ser modificado a nuevas necesidades.

Este artefacto se muestra en la siguiente página.

Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software



Estándar para modificar el proceso

El propósito de este artefacto es describir la forma en la que debe ser modificado el proceso con la finalidad de mejorarlo y adaptarlo a los cambios del medio ambiente.

Comité responsable del proceso

Se debe definir un equipo de personas responsables de analizar, evaluar y hacer los cambios en el proceso.

Cambios en el proceso

Durante el seguimiento de actividades y pasos los involucrados deben tener presente que el proceso es susceptible de ser modificado para adecuarlo a nuevas necesidades o corregir detalles en actividades, pasos o artefactos.

Cuando alguna persona tiene una sugerencia sobre algún cambio en el proceso deberá de llenar el artefacto "Solicitud de cambio al PDSIIB" y deberá entregarlo al coordinador del comité responsable del proceso.

Cuando exista una solicitud de cambio al PDSIIB el coordinador deberá de reunir al comité para hacer un análisis y evaluación para determinar si debe llevarse a cabo el cambio.

Se debe actualizar el artefacto "Lista de actividades, artefactos y roles" del proceso de desarrollo que se encuentra en la actividad "Conocer el ambiente para el proyecto".

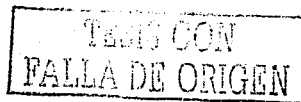
Si el cambio se va a realizar se debe indicar quienes serán los responsables.

Biblioteca de componentes

Cuando una persona, desarrolla un componente, una función, un procedimiento, un módulo, etc., donde se tenga una buena perspectiva de reutilización para el desarrollo de nuevos sistemas se deberá hacer la notificación al responsable de la actualización de la biblioteca de componentes, dicho responsable se encuentra registrado en el artefacto llamado "Comité responsable del PDSIIB".

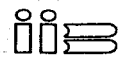
Al responsable de actualizar la biblioteca de componentes se le debe entregar el código fuente del componente, función, procedimiento o módulo con la descripción, los parámetros de entrada, parámetros de salida, lenguaje, plataforma y un ejemplo de cómo se utiliza.

La Biblioteca de componentes se encuentra en el artefacto "Estándar de herramientas".



**TESIS CON
FALLA DE ORIGEN**

**Instituto de Investigaciones Bibliográficas
Departamento de Informática
Sección de desarrollo de Software**



Comité responsable del PDSIIB

Responsables de analizar, evaluar y modificar el proceso de desarrollo de sistemas.

	Firma
Coordinador _____	_____
Responsable _____	_____
Responsable _____	_____
Responsable _____	_____

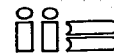
Fecha: _____

Responsable de actualizar la biblioteca de componentes

	Firma
Responsable _____	_____
Suplente _____	_____

Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software



Solicitud de cambio al PDSIIB

Nombre de quien solicita el cambio _____

Elemento a modificar

Artefacto _____

Actividad _____

Paso _____

Otro _____

Explicación

Elemento a agregar

Artefacto _____

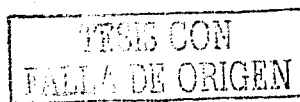
Actividad _____

Paso _____

Otro _____

Explicación

Fecha _____



PDSIIB <Artefacto> Estándar de herramientas

En este artefacto se listan las herramientas con las que cuenta el Departamento de Informática para el desarrollo de sistemas, se listan los nombres y versiones de los sistemas operativos, manejadores de bases de datos, lenguajes de programación, biblioteca de componentes, servidores de páginas web y otras.

La finalidad de tener este estándar es conocer las herramientas que se tienen para el desarrollo de sistemas y saber que plataformas se deben considerar para la configuración.

Este artefacto se muestra en la siguiente página.

Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software



Estándar de herramientas

El propósito de este artefacto es señalar las herramientas disponibles para el proceso de desarrollo de sistemas.

1. Herramientas para requerimientos

Las herramientas que se listan enseguida deben ser consideradas antes de comenzar la fase de inicio.

Sistemas operativos	
Nombre	Versión
Solaris	5.0, 8.0
Linux Mandrake	9.0
Netware	3.12
Windows NT	3.5, 4.0
Windows	3.11, 95, 98, 2000, XP
MS-DOS	6.2
Incluir nuevas adquisiciones	

2. Herramientas para análisis y diseño

Las herramientas que se listan enseguida deben ser consideradas antes de comenzar la fase de elaboración.

Varias	
Nombre	Versión
Word	97, 2000
Excel	97, 2000
Incluir nuevas adquisiciones	

3. Herramientas para implementación y pruebas

Las herramientas que se listan enseguida deben ser consideradas antes de comenzar la fase de construcción.

Lenguajes de programación	
Nombre	Versión
Visual Basic	4.0, 6.0
Java	Jdk 1.3
Incluir nuevas adquisiciones	

TESIS CON
FALLA DE ORIGEN

**TESIS CON
FALLA DE ORIGEN**

Manejadores de base de datos	
Nombre	Versión
Oracle	8i
Postgresql	7.2
Microsoft Access	2.0, 97, 2000
Microisis	3.03
Winisis	1.4
Incluir nuevas adquisiciones	

Servidores de páginas web	
Nombre	Versión
Apache	1.3, 2.0
Tomcat	4.0
Incluir nuevas adquisiciones	

Reporteadores	
Nombre	Versión
Crystal reports	3.0.1.32
Incluir nuevas adquisiciones	

Biblioteca de componentes		
Nombre	Versión	
Isisdll	4.0	
NombreDeFunción	Descripción de la función	
	Parámetros de entrada y salida:	
	Lenguaje y plataforma	
	Ejemplo de cómo se utiliza	
Incluir nuevas componentes, funciones o procedimientos		

4. Procedimientos de instalación de herramientas
Aquí se coloca el procedimiento para realizar la instalación de las herramientas.
5. Procedimientos de configuración de herramientas
Aquí se coloca el procedimiento para realizar la configuración de las herramientas.
6. Procedimientos de verificación de herramientas
Aquí se coloca el procedimiento para realizar la verificación de herramientas instaladas.

PDSIIB <Artefacto> Herramientas a comprar

En este artefacto se listan los nombres de las herramientas que se van a comprar para el proyecto, se indica la versión y en qué sistema operativo opera.

Con este artefacto se hace la petición formal al jefe del Departamento de Informática para que las herramientas se compren.

Cuando las herramientas ya han sido compradas se deben instalar y después se debe avisar a todos los involucrados en el desarrollo del sistema.

Las herramientas que se han comprado se deben registrar en el artefacto llamado "Estándar de herramientas", y cuando es necesario, se debe hacer la actualización en la sección 6. Procedimientos de instalación de herramientas, en la sección 7. Procedimientos de configuración de herramientas y en la sección 8. Procedimientos de verificación de herramientas.

Este artefacto se muestra en la siguiente página.

TESIS CON
FALLA DE ORIGEN

TESTS CON
FALLA DE ORIGEN



Instituto de Investigaciones Bibliográficas
Departamento de Informática
Sección de desarrollo de Software

Herramientas a comprar

Siglas del sistema _____ Ciclo _____
Responsable _____ Fecha _____

Paso: Identificar las herramientas a comprar

Jefe del Departamento de Informática
Presente

Por este conducto le solicito a usted sea tan amable de realizar las gestiones necesarias para comprar las herramientas que se listan a continuación.

Nombre de la herramienta	Versión	Sistema operativo donde opera

Estas herramientas se utilizarán para realizar el desarrollo del [nombre del sistema].

Sin más por el momento me despido de usted enviándole un cordial saludo.

Atentamente

Nombre
Especialista en herramientas

PDSIIB <Artefacto> Configuración de herramientas

En este artefacto se indican las acciones requeridas sobre la configuración de herramientas que serán utilizadas en un ciclo, indicando el nombre de la herramienta y las acciones a realizar para hacer la configuración.

La finalidad de este artefacto es actuar como una memoria donde en cualquier momento se pueda conocer la configuración de alguna herramienta. Tener especificada la configuración de dichas herramientas ayudará a considerar la revisión de la configuración cuando ocurren fallas en el sistema. También actuará como un punto importante a considerar cuando se realizan las pruebas.

Este artefacto se muestra en la siguiente página.

TESIS CON
FALLA DE ORIGEN

TESIS CON
FALLA DE ORIGEN



Instituto de Investigaciones Bibliográficas
Departamento de Informática
Sección de desarrollo de Software

Configuración de herramientas

Siglas del sistema _____ Ciclo _____
Responsable _____ Fecha _____

Paso: Indicar la configuración de las herramientas (Revisar el "Estándar de herramientas")

Nombre de la herramienta	Configuración	Verificado

Vo. Bo.

Nombre
Administrador del proyecto

Capítulo 4

Flujo de trabajo distribución

El propósito de este flujo es asegurar que el sistema estará disponible para los usuarios finales. La descripción a detalle de acuerdo al Proceso Unificado de Rational se encuentra en el Apéndice B.

En este capítulo se explica como es el flujo de trabajo distribución, las consideraciones para hacer la adaptación y el resultado de la adaptación de cada actividad y sus pasos. Se hace una explicación de los artefactos y para los roles se hace una descripción y se describe el perfil.

Para explicar cada actividad se tiene el siguiente formato: se pone del lado izquierdo la etiqueta RUP, el nombre de la actividad y el nombre del rol responsable de la actividad, de acuerdo a como lo establece RUP. Después se pone la etiqueta ADAPTACION y el nombre de la actividad y enseguida las consideraciones para hacer la adaptación. Después se utiliza la etiqueta PDSIIB para designar al proceso de desarrollo de sistemas del Instituto de Investigaciones Bibliográficas en el Departamento de Informática, se pone el nombre de la actividad como quedó en el proceso resultante y el nombre del rol responsable. Cuando finaliza la descripción de la actividad se pone a nivel de pasos cada actividad con la etiqueta PDSIIB y el nombre del paso con su descripción.

Actividades

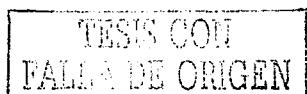
RUP <Actividad> Desarrollar el plan de distribución
<Rol> Administrador de distribución

Los propósitos de esta actividad son especificar las necesidades para entregar el producto al usuario final. El plan de distribución se enfoca a la entrega del sistema junto con la elaboración del material de apoyo y capacitación para asegurar que los usuarios finales puedan usarlo.

El plan de distribución incluye los siguientes planes: el plan para producir el software, el plan para empaquetar el software, el plan para distribuir el software, el plan para instalar el software, la migración de datos, la forma de capacitar al usuario y también incluye la emisión de la factura.

ADAPTACIÓN <Actividad> Desarrollar el plan de distribución

Esta actividad integra en un sólo plan de distribución algunos de los aspectos importantes de cada uno de los planes que lo integran, los datos que se registran en este plan son las acciones que debe realizar el usuario responsable, se listan las herramientas que acompañan al sistema, se define la forma en que se realizará la migración de datos, se listan los recursos materiales para hacer la distribución, se registran los casos de uso que





serán entregados, los manuales que acompañan al sistema y se define la forma en que se llevará a cabo la capacitación de los usuarios finales.

El departamento no hace sistemas para venderlos, los que se desarrollan se instalan internamente, por ello, no se emiten facturas.

PDSIIB <Actividad> Hacer plan de distribución
<Rol> Administrador de distribución

En el plan de distribución se describen las necesidades en cuanto a recursos necesarios para hacer la entrega del sistema, herramientas que lo acompañan, se define la forma para hacer la migración de datos, que manuales de usuarios se entregarán y como será la capacitación de usuarios.

RUP <Actividad> Desarrollar el material de apoyo
<Rol> Escritor técnico

El propósito de esta actividad es producir el material necesario para capacitar a los usuarios.

Se hace un bosquejo de los materiales determinando el público a capacitar, se considera el tipo de roles de las personas típicas de la organización y se definen los objetivos de la capacitación.

La capacitación puede ser apoyada con tutoriales en línea. Incluir capacitación en salones de clase y con talleres. Se debe decidir la duración de los cursos y el número de participantes.

Para asegurar una buena calidad de los materiales puede involucrar a un asesor para repasar los materiales y discutir problemas particulares. Se podría incluir a los representantes de los usuarios finales y a desarrolladores del producto.

Los materiales de apoyo para los usuarios finales también pueden estar contenidos en sitios web. Producir este tipo de materiales es en sí mismo un proyecto que puede manejarse en iteraciones.

ADAPTACIÓN <Actividad> Desarrollar el material de apoyo

Esta actividad cambió un poco, ya que no se tiene personal suficiente para dedicarle tiempo a crear materiales de apoyo para la capacitación de los usuarios. Por ello, el material que se utilizará para los cursos de capacitación son: el manual de configuración, el manual de instalación y el manual de usuario.

PDSIIB <Actividad> Hacer manuales
<Rol> Escritor técnico

Consiste en elaborar el manual de configuración, el manual de instalación, el manual de usuario y en conformar el manual técnico.

El manual técnico se conforma por los artefactos generados en el proyecto.

RUP <Actividad> Administrar las pruebas de aceptación
<Rol> Administrador de distribución, probador

El propósito de esta actividad es asegurar que el producto cumple con los criterios de aceptación en el sitio de desarrollo y en el sitio de instalación designado. Se realizan en el sitio de desarrollo para después hacerlas en producción.

El administrador de distribución debe quedar satisfecho con respecto a la disposición de todos los elementos que serán probados.

El probador carga todo el software y procede con la ejecución de los casos de prueba.

En caso de que los resultados de las pruebas sean considerados como inaceptables el administrador de distribución puede hacer solicitudes de cambio.

ADAPTACIÓN <Actividad> Administrar las pruebas de aceptación

A esta actividad se agregaron pasos para preparar el artefacto "Invitación de pruebas formales". Después de que se verifica la disponibilidad para hacer las pruebas, se prepara la invitación y se hace entrega de dicha invitación indicando la fecha, la hora y el lugar donde se realizarán las pruebas.

Después se realizan las pruebas y la evaluación de las mismas, estas dos actividades se realizan en el flujo de trabajo pruebas.

PDSIIB <Actividad> Realizar pruebas de aceptación
<Rol> Administrador de distribución, probador

Consiste en probar el software en el sitio de desarrollo para asegurar que el sistema cumple con los criterios de aceptación para después hacerlas en el sitio de producción.

En el PDSIIB existen las actividades "Realizar pruebas de aceptación en desarrollo" y "Realizar pruebas de aceptación en sitio", la diferencia es que la primera se realiza en el Departamento de Informática y la segunda se realiza en el lugar de trabajo del usuario, es decir, en producción.

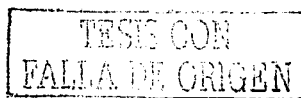
RUP <Actividad> Producir la unidad de distribución
<Rol> Administrador de distribución

Esta actividad tiene el propósito de generar la unidad de distribución con material de apoyo para el usuario final, notas de liberación y artefactos de instalación.

La unidad de distribución contiene todos los elementos entregables y éstos se listan en la factura de materiales.

ADAPTACIÓN <Actividad> Producir la unidad de distribución

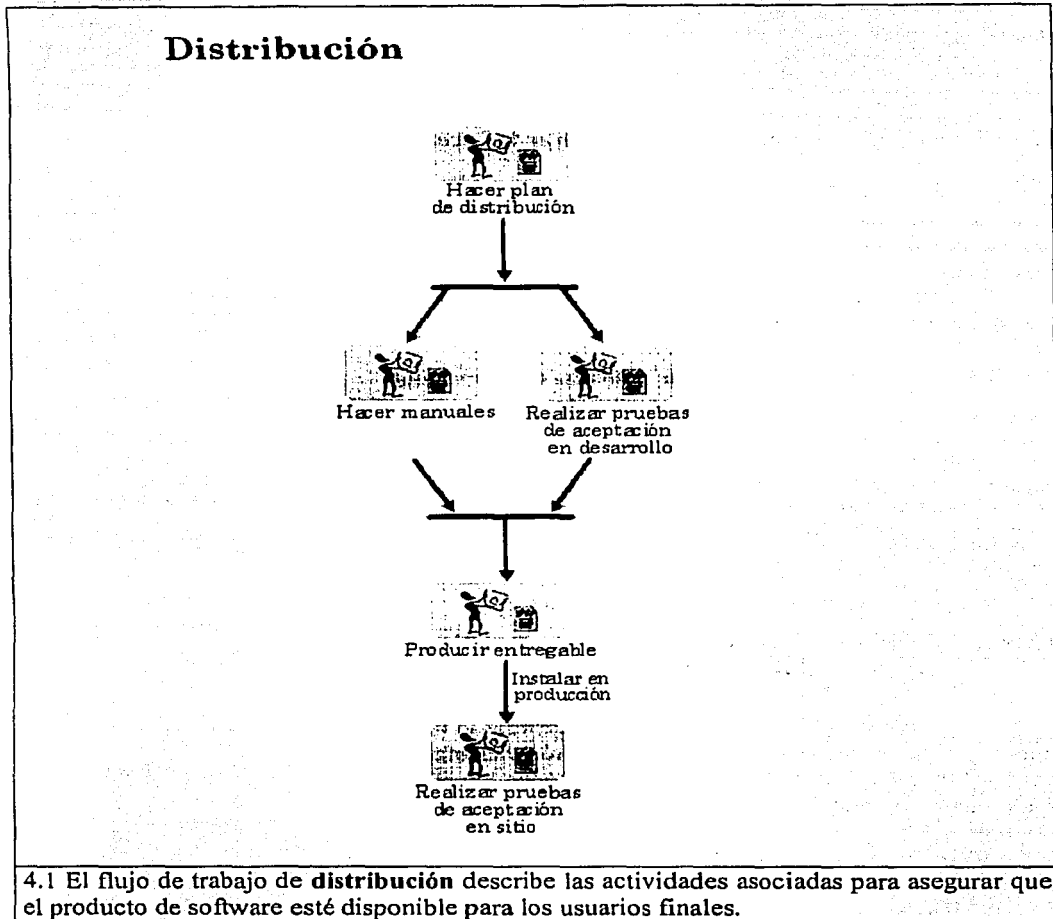
Esta actividad cambió debido a que los manuales se generan en la actividad de "hacer manuales", aquí únicamente se reúnen los manuales y herramientas que se entregarán a los usuarios finales.



PDSIIB <Actividad> **Producir entregable**
<Rol> **Administrador de distribución**

Consiste en indicar los manuales y herramientas que acompañan al sistema que serán entregados a los usuarios finales.

El nuevo flujo de trabajo **distribución** se puede apreciar en el diagrama 4.1.



PDSIIB <Actividad> Hacer plan de distribución
<Rol> Administrador de distribución

PDSIIB <Paso> Reunir artefactos de entrada

Reunir los artefactos de entrada necesarios correspondientes a esta actividad. Los artefactos son: Plan de desarrollo de software, plan de ciclo, estándar de distribución y el modelo de diseño.

PDSIIB <Paso> Definir acciones que debe hacer el usuario responsable

Consiste en registrar en el plan de distribución, el hardware, el software, la red y otros recursos necesarios para que los usuarios finales puedan utilizar el sistema que será entregado.

Estos datos se obtienen del artefacto "Visión inicial".

PDSIIB <Paso> Definir herramientas que acompañan al sistema

Consiste en registrar en el plan de distribución que herramientas de software adicional debe de entregarse a los usuarios finales para que puedan ejecutar el sistema.

Estos datos se obtienen del artefacto "Modelo de diseño".

PDSIIB <Paso> Definir recursos materiales para la distribución

Consiste en registrar en el plan de distribución los recursos materiales necesarios para poder hacer la distribución del sistema, imprimir manuales y colocar las herramientas que acompañan al sistema en algún medio magnético.

Se debe consultar el artefacto "Estándar de distribución".

PDSIIB <Paso> Definir migración de datos

Consiste en registrar en el plan de distribución la forma en que se va a realizar la exportación e importación de datos para el sistema que se está desarrollando.

Se debe consultar el artefacto "Estándar de distribución" en la sección "Migración de datos".

PDSIIB <Paso> Listar casos de uso a entregar

Consiste en registrar en el plan de distribución los nombres de los casos de uso o módulos que serán entregados. Los casos de uso a entregar se obtienen del artefacto "Plan de ciclo".

Cuando se va a desarrollar un programa para hacer la migración de datos, se agrega un ciclo más al proyecto.

Se hace la lista de los casos de uso cuando hay entregas parciales del sistema, y cuando se hace la entrega de todos los casos de uso únicamente se pone el nombre del sistema.

PDSIIB <Paso> Listar documentación a liberar

Consiste en indicar en el plan de distribución que manuales se van a elaborar.

Los manuales se entregan cuando el sistema esta completo, sin embargo, puede ocurrir por necesidades del usuario que se hagan entregas parciales de los manuales.

Es recomendable hacer los manuales por cada ciclo para evitar carga de trabajo en las etapas finales del proyecto.

PDSIIB <Paso> Definir forma de capacitar al usuario

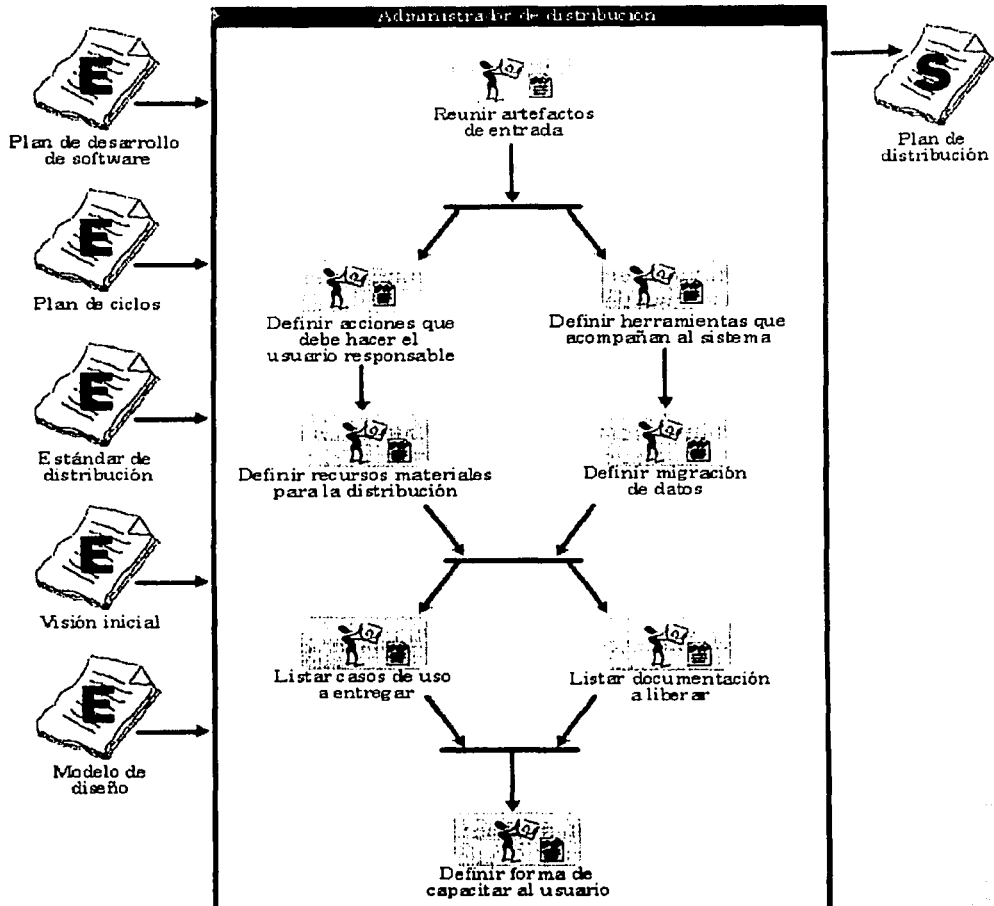
Consiste en indicar en el plan de distribución cómo se va a realizar la capacitación de los usuarios y en que lugar. Los datos se obtienen del artefacto "Visión inicial".

Artefactos de salida

Plan de distribución.

La actividad **hacer plan de distribución** se puede apreciar en el diagrama 4.2.

Hacer plan de distribución



4.2 Diagrama de actividades hacer plan de distribución

PDSIIB <Actividad> Hacer manuales
<Rol> Escritor técnico

PDSIIB <Paso> Reunir artefactos de entrada

Reunir los artefactos de entrada necesarios correspondientes a esta actividad. Los artefactos son: Modelo de análisis, modelo de diseño, modelo de implementación, estándar de distribución, sistema y la configuración del sistema.

PDSIIB <Paso> Hacer manual de configuración

Consiste en escribir las instrucciones de configuración tomando los datos del artefacto "Configuración del sistema".

Escribir las instrucciones de configuración a través de secuencias de pasos específicos fáciles de seguir. Debe incluir información suficiente para los novatos. Las explicaciones deben ser apoyadas con ejemplos.

Incluir imágenes de las pantallas que aparecen cuando se hace la configuración.

Se deben seguir las instrucciones que se escribieron en el manual para verificar que son correctas.

Consultar el artefacto "Estándar de distribución" en la sección "Estándar para los manuales del sistema".

El manual de configuración no se imprime en este paso.

PDSIIB <Paso> Hacer manual de instalación

Consiste en escribir las instrucciones de instalación. Escribir las instrucciones de instalación a través de secuencias de pasos específicos fáciles de seguir. Debe incluir información suficiente para los novatos. Las explicaciones deben ser apoyadas con ejemplos.

Incluir imágenes de las pantallas que aparecen cuando se hace la instalación.

Se deben seguir las instrucciones que se escribieron en el manual para verificar que son correctas.

Consultar el artefacto "Estándar de distribución" en la sección "Estándar para los manuales del sistema".

El manual de instalación no se imprime en este paso.

PDSIIB <Paso> Hacer manual de usuario

Consiste en escribir las instrucciones para que el sistema funcione de acuerdo a los casos de uso o módulos que lo integran. Los datos se obtienen del modelo de diseño y de las pantallas que se capturan del sistema.

Escribir las instrucciones del sistema a través de secuencias de pasos específicos fáciles de seguir. Las explicaciones deben ser apoyadas con ejemplos.

Incluir imágenes de las pantallas que aparecen cuando se hacen los pasos para ejecutar las instrucciones.

Se deben seguir las instrucciones que se escribieron en el manual para verificar que son correctas.

Incluir el índice del manual.

Consultar el artefacto "Estándar de distribución" en la sección "Estándar para los manuales del sistema".

El manual de usuario no se imprime en este paso.

PDSIIB <Paso> Hacer manual técnico

Consiste en conformar el manual técnico con los artefactos generados en el proyecto.

Hacer una carátula con la etiqueta de "Manual técnico", el nombre del sistema y versión y el índice de contenido.

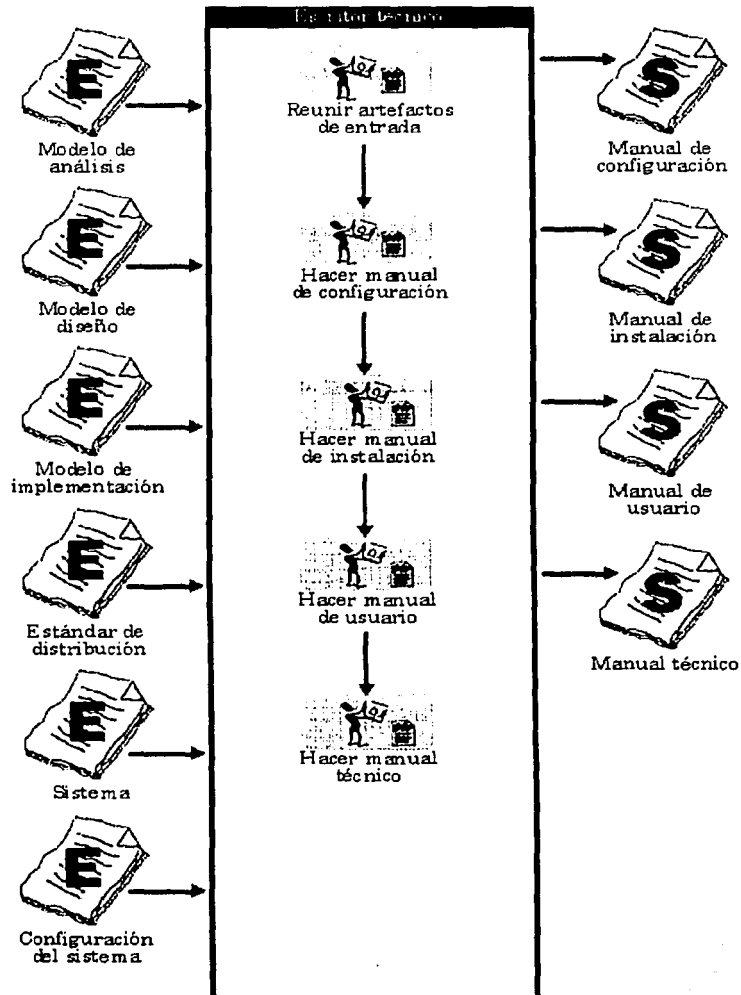
El manual técnico no se imprime en este paso.

Artefactos de salida

Manual de configuración, manual de instalación, manual de usuario y manual técnico.

La actividad **hacer manuales** se puede apreciar en el diagrama 4.3.

Hacer manuales



4.3 Diagrama de actividades hacer manuales

PDSIIB <Actividad> Realizar pruebas de aceptación en desarrollo
<Rol> Administrador de distribución, probador

PDSIIB <Paso> Reunir artefactos de entrada

Reunir los artefactos de entrada necesarios correspondientes a esta actividad. Los artefactos son: Plan de pruebas, sistema, casos de prueba.

PDSIIB <Paso> Verificar disponibilidad para hacer pruebas formales

Consiste en revisar que la red, el sistema operativo, el manejador de base de datos, los servidores y el sistema están listos para hacer las pruebas.

Revisar que el sistema cumple con los requerimientos y que están listos los artefactos a entregar.

Revisar el plan de pruebas.

PDSIIB <Paso> Preparar invitación a los interesados de las pruebas

Consiste en llenar el artefacto "Invitación de pruebas formales" especificando la fecha, la hora, el nombre del sistema y el lugar en donde se realizarán las pruebas, la lista de los casos de uso o módulos y los nombres de los usuarios a invitar.

Los nombres de los casos de uso o módulos a entregar se obtienen de los artefactos "Plan de ciclo" y "Cierre de ciclo".

La "Invitación de pruebas formales" se hace una vez que se ha convenido con la mayoría de los interesados de una manera informal, ahora se va a formalizar.

Para las entregas de ciclo se listan los nombres de los casos de uso y cuando la liberación es del sistema completo únicamente se pone el nombre del sistema.

Llenar el artefacto "Aceptación de pruebas en desarrollo", en éste se registra el nombre del sistema, los casos de uso o módulos a liberar y la firma del usuario responsable.

Para las entregas parciales se listan los nombres de los casos de uso y si la liberación es de todos los casos de uso, únicamente se pone el nombre del sistema.

PDSIIB <Paso> Entregar invitación de las pruebas formales

Consiste en repartir las invitaciones a los interesados en hacer las pruebas del sistema.

Entregar la invitación a todos los usuarios invitados sin que falte nadie.

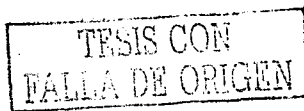
Enviar correo electrónico con solicitud de confirmación de que ha sido leído cuando las invitaciones se hacen electrónicamente.

PDSIIB <Paso> Recibir a los interesados de las pruebas formales

Consiste en dar la bienvenida a los asistentes y dar una breve explicación de los objetivos de las pruebas formales.

Precisar la importancia de las pruebas.

La explicación no debe tardar mas de diez minutos.



PDSIIB <Paso> Ejecutar las pruebas

Consiste en llevar a cabo el procedimiento de los casos de prueba registrando los resultados de las pruebas en el artefacto "Caso de prueba".

PDSIIB <Paso> Evaluar las pruebas

Consiste en estimar la calidad del sistema, hacer solicitudes de cambio y establecer la cobertura de las pruebas.

PDSIIB <Paso> Preparar aceptación de pruebas en desarrollo

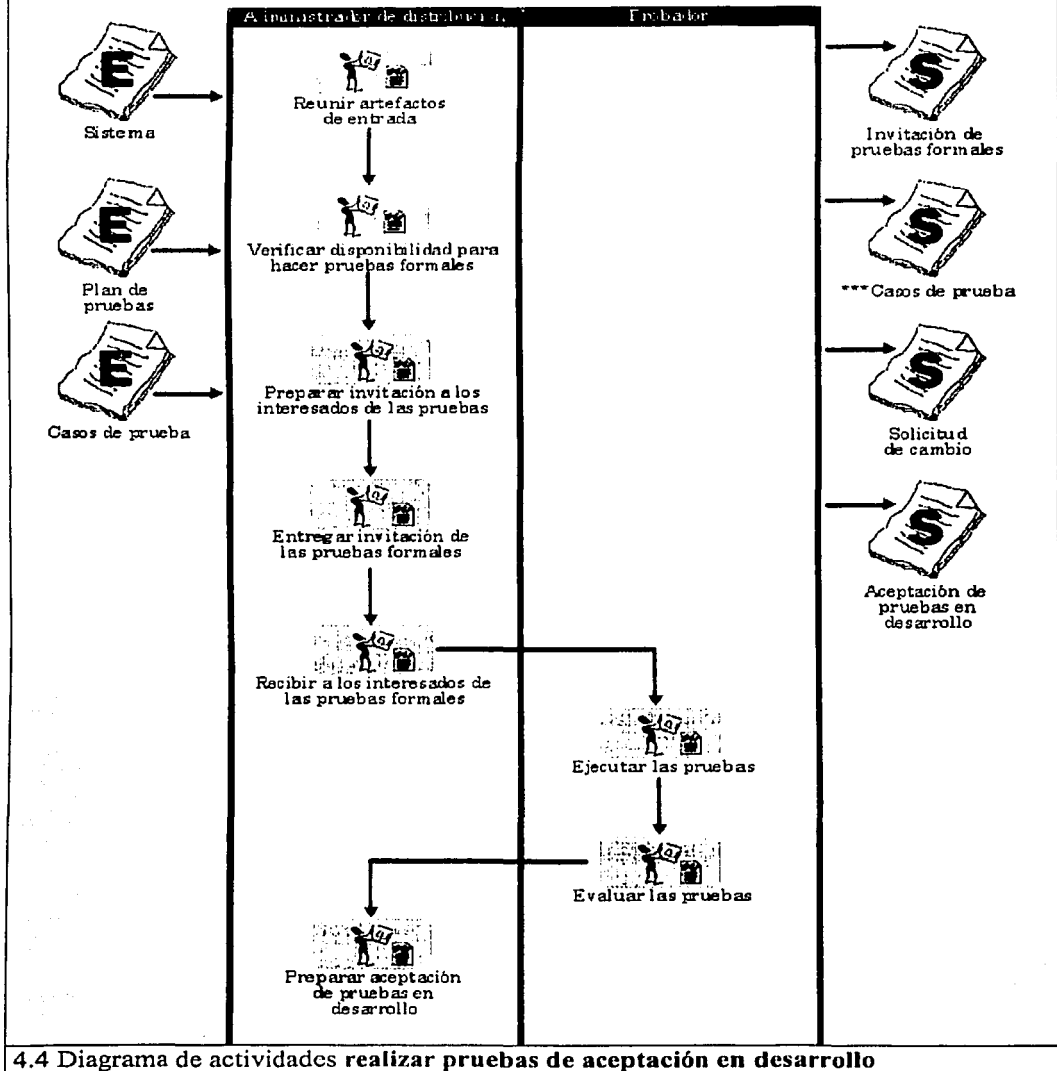
Consiste en comunicar al usuario responsable que tiene que firmar el artefacto "Aceptación de pruebas en desarrollo". Esto sucede cuando no hubo ningún problema al hacer las pruebas.

Artefactos de salida

Invitación de pruebas formales, casos de prueba, solicitud de cambio, aceptación de pruebas en desarrollo.

La actividad **realizar pruebas de aceptación en desarrollo** se puede apreciar en el diagrama 4.4.

Realizar pruebas de aceptación en desarrollo



TESIS CON
FALLA DE ORIGEN

PDSIIB <Actividad> Producir entregable
<Rol> Administrador de distribución

PDSIIB <Paso> Reunir artefactos de entrada

Reunir los artefactos de entrada necesarios correspondientes a esta actividad. Los artefactos son: Sistema, herramientas que acompañan al sistema, manuales, configuración del sistema y cierre de ciclo.

PDSIIB <Paso> Reunir manuales a entregar

Consiste en tener listos los manuales de configuración, de instalación y de usuario del sistema.

También se debe tener listo el manual técnico.

Se indica en el artefacto "Informe de entregable" los manuales a entregar.

PDSIIB <Paso> Reunir las herramientas que acompaña al sistema

Consiste en tener listo todo el software adicional para que el sistema pueda ser instalado. Los nombres de las herramientas se registran en el artefacto "Informe de entregable". Los datos se obtienen del plan de distribución.

PDSIIB <Paso> Realizar especificaciones del sistema en internet

Consiste en indicar la dirección electrónica donde se encuentra el nuevo sistema. Se llena únicamente cuando el sistema se ha desarrollado para acceso desde un visualizador de web. Estos datos se registran en la sección especificaciones del sistema en internet del artefacto "Informe de entregable".

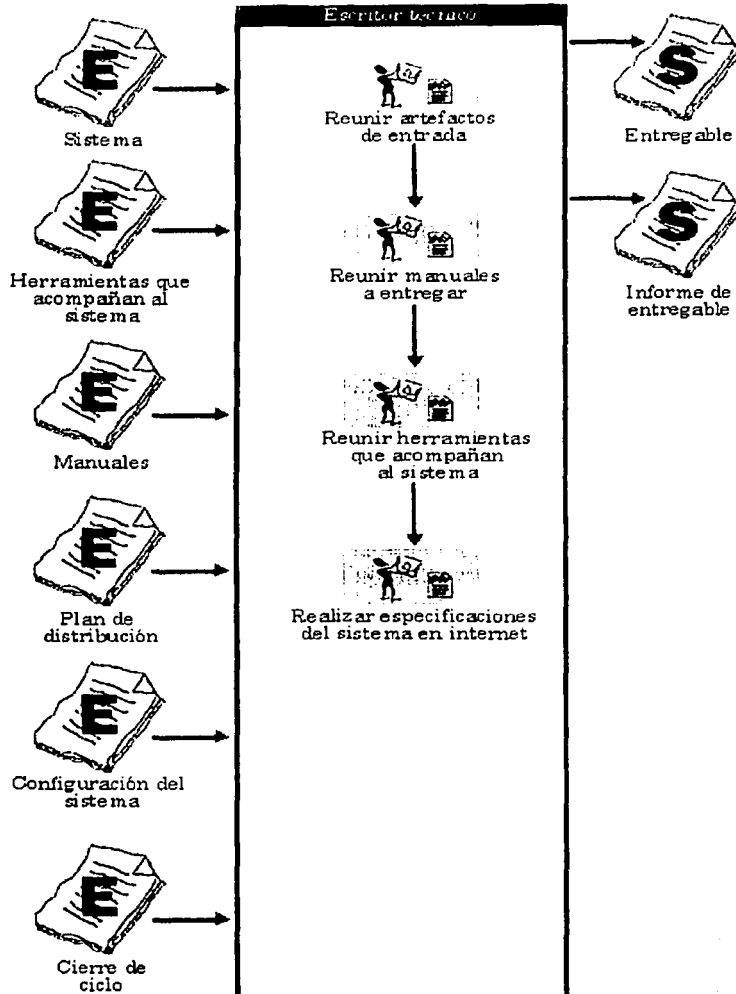
Los datos se obtienen del artefacto "Configuración del sistema".

Artefactos de salida

Entregable, informe de entregable.

La actividad **producir entregable** se puede apreciar en el diagrama 4.5.

Producir entregable



4.5 Diagrama de actividades producir entregable

TESIS CON
FALLA DE ORIGEN

PDSIIB <Actividad> Realizar pruebas de aceptación en sitio
<Rol> Administrador de distribución, probador

PDSIIB <Paso> Reunir artefactos de entrada

Reunir los artefactos de entrada necesarios correspondientes a esta actividad. Los artefactos son: Plan de pruebas y casos de prueba.

PDSIIB <Paso> Verificar disponibilidad para hacer pruebas formales

Consiste en revisar que la infraestructura de hardware y software está lista para hacer las pruebas en producción.

Revisar que el sistema desarrollado cumple con todos los requerimientos y los artefactos necesarios que serán entregados.

Revisar que todo el software adicional se encuentra instalado correctamente, sistema operativo, servidores, etc.

Revisar que la red está disponible y configurada de acuerdo a las especificaciones.

PDSIIB <Paso> Preparar invitación a los interesados de las pruebas

Consiste en registrar en el artefacto "Invitación de pruebas formales" el nombre del usuario responsable, el nombre del sistema, la fecha, la hora, el lugar en donde se realizarán las pruebas, la lista de los casos de uso o módulos y los nombres de los usuarios invitados.

Los nombres de los casos de uso o módulos a entregar se obtienen de los artefactos "Plan de ciclo" y "Cierre de ciclo".

La "Invitación de pruebas formales" se hace una vez que se ha convenido con la mayoría de los interesados de una manera informal, ahora se va a formalizar.

Llenar el artefacto "Aceptación de pruebas en sitio", en éste se registra el nombre del sistema, los casos de uso o módulos a liberar y la firma del usuario responsable.

Para las entregas parciales se listan los nombres de los casos de uso y si la liberación es de todos los casos de uso, únicamente se pone el nombre del sistema.

PDSIIB <Paso> Entregar invitación de las pruebas formales

Consiste en repartir las invitaciones a los interesados del sistema. Entregar la invitación a todos los interesados sin que falte nadie.

Enviar correo electrónico con solicitud de confirmación de que ha sido leído cuando las invitaciones se hacen electrónicamente.

PDSIIB <Paso> Recibir a los interesados de las pruebas formales

Consiste en dar la bienvenida a los asistentes y dar una breve explicación de los objetivos de las pruebas formales.

Precisar la importancia de las pruebas.

La explicación no debe tardar mas de diez minutos.

PDSIIB <Paso> Ejecutar las pruebas

Consiste en llevar a cabo el procedimiento de los casos de prueba registrando los resultados de las pruebas en el artefacto "Caso de prueba".

PDSIIB <Paso> Evaluar las pruebas

Consiste en estimar la calidad del sistema, hacer solicitudes de cambio y establecer la cobertura de las pruebas.

PDSIIB <Paso> Pasar entregable al administrador del proyecto

Consiste en hacer la entrega del sistema, los manuales y las herramientas que acompañan al sistema al administrador del proyecto. Junto con lo anterior se entrega el artefacto "Aceptación de pruebas en sitio" y el "Informe de entregable".

Preparar la "Aceptación de pruebas en sitio" y el "Informe de entregable" con anticipación para evitar demoras.

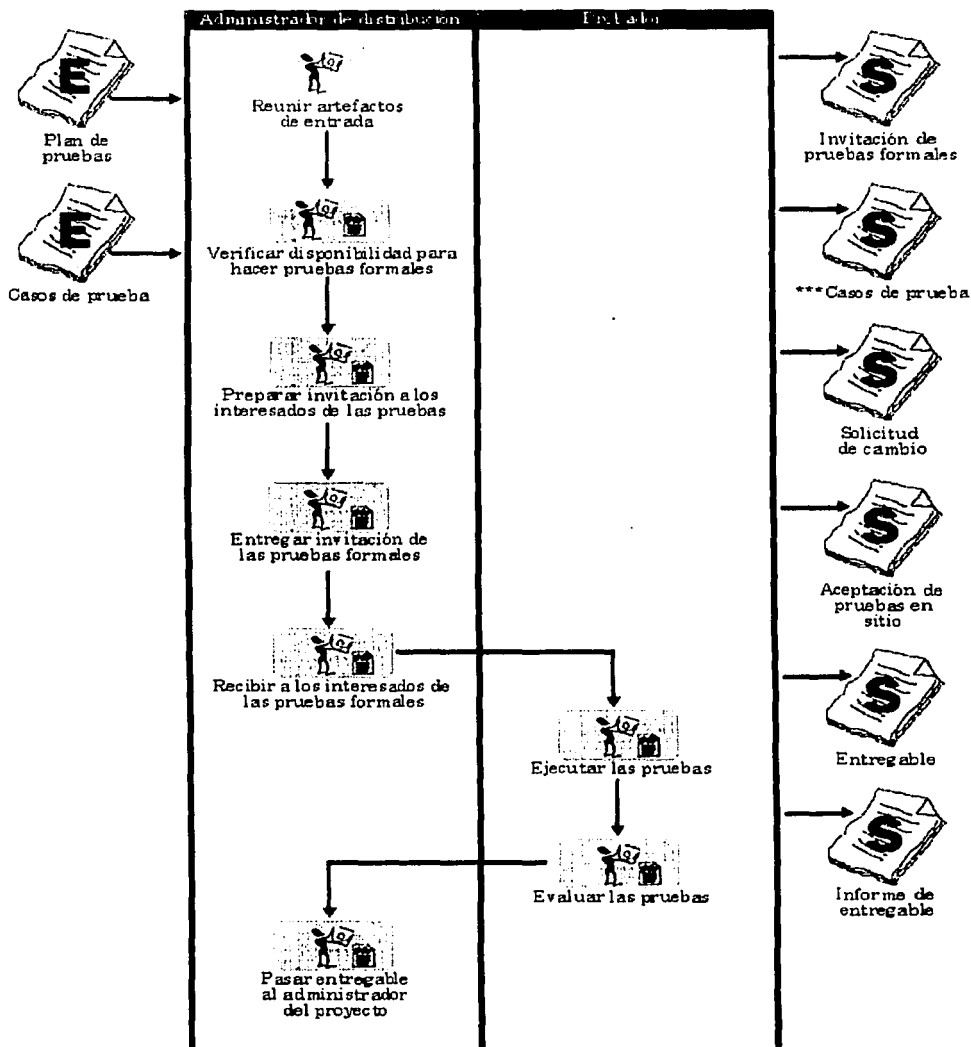
Artefactos de salida

Invitación de pruebas formales, casos de prueba, solicitud de cambio, aceptación de pruebas en sitio, entregable y el informe de entregable.

La actividad **realizar pruebas de aceptación en sitio** se puede apreciar en el diagrama 4.6.

TEMAS CON FALLA DE ORIGEN

Realizar pruebas de aceptación en sitio



4.6 Diagrama de actividades realizar pruebas de aceptación en sitio

Roles

PDSIIB <Rol> Administrador de distribución

Es el encargado de planear la transición del sistema a la comunidad de usuarios utilizando varios artefactos asociados.

Perfil del rol

Debe tener experiencia en distribución de sistemas, habilidad para comunicar las necesidades de las actividades de distribución al resto de la organización, habilidad para planear.

PDSIIB <Rol> Escritor técnico

Es el encargado de hacer el material de apoyo para usuarios finales como manuales de usuario, guías, textos de ayuda, notas de liberación, etc.

Perfil del rol

Debe tener experiencia en escritura técnica, sistemas de ayuda, sitios web de ayuda y habilidades de comunicación.

TESIS CON
FALLA DE ORIGEN

Artefactos

PDSIIB <Artefacto> Estándar de distribución

En este artefacto se describen los aspectos necesarios que se deben tomar en cuenta para obtener los recursos materiales necesarios para que el sistema y artefactos adicionales sean entregados al usuario final.

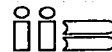
También se indica a que departamentos del Instituto se debe hacer la petición de recursos, se definen las formas de capacitación, se establecen los aspectos a tomar en cuenta para hacer la migración de datos.

Este estándar también incluye el "Estándar para los manuales del sistema", donde se especifica el formato que deben tener los manuales y las reglas que deben cumplir para su aceptación.

Este artefacto se muestra en la siguiente página.

Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software



Estándar de distribución

El propósito de este artefacto es establecer los aspectos necesarios que se deben tomar en cuenta para obtener los recursos materiales necesarios para que el sistema y artefactos adicionales sean entregados al usuario final.

Definiciones

Entregable. Es el sistema con las funcionalidades establecidas para un ciclo de desarrollo, artefactos adicionales y herramientas de software que acompañan al sistema.

Recursos: Son todo tipo de material indispensable para apoyar el desarrollo de un sistema y sus artefactos.

Herramientas que acompañan al sistema: Son todo tipo de software necesarios para que el sistema pueda ejecutarse de acuerdo a los requerimientos especificados.

Lista de recursos utilizados frecuentemente

Hardware: Computadoras, impresoras, reguladores, no-breaks, concentradores, etc.

Software: Sistemas operativos, manejadores de base de datos, lenguajes de programación, bibliotecas de objetos o funciones, reporteadores, etc.

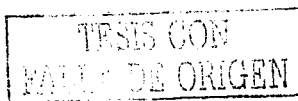
Consumibles: Discos compactos, discos flexibles, toners y cartuchos de impresora, etc.

Papelería: Lápices, plumas, gomas, hojas, sacapuntas, engrapadoras, etiquetas, etc.

Libros y manuales: Libros sobre herramientas de software necesario para hacer el desarrollo de un sistema o manuales de usuario, manual de configuración, manual de instalación y todo tipo de manuales que vienen con el software.

Petición de recursos por parte del administrador de distribución

- Cuando el recurso es del tipo hardware o software la petición se hace al Departamento de Informática.
- Cuando el recurso es del tipo consumible o papelería la petición se hace al Departamento de Suministros del Instituto.
- Cuando el recurso es sobre libros o manuales la petición se hace al departamento de adquisiciones de la Biblioteca Nacional.



Petición de recursos por parte del usuario responsable

- Cuando el recurso es del tipo hardware o software la petición se hace a la institución que apoya el proyecto de investigación.
- Cuando el sistema a desarrollar no tiene ningún tipo de apoyo de instituciones externas el recurso de hardware y software se deben solicitar al Departamento de Informática.
- Cuando el recurso es un nodo de red se debe solicitar a la Secretaría Administrativa realice las gestiones necesarias para instalar el cableado en el lugar requerido. Posteriormente se solicita al Departamento de Informática la conexión de la placa.

Formas de capacitación

- Capacitación a un usuario responsable

Se capacita a un usuario para que él después haga lo mismo con los demás usuarios del sistema. Esta es la forma determinada para hacer la capacitación, sin embargo, cuando el usuario responsable considere por diversas razones que se debe capacitar a los demás usuarios se toman en consideración las otras formas de capacitación.

- Capacitación a todos los usuarios

Esta forma se utiliza cuando por la complejidad del sistema se requieren explicaciones más precisas sobre como se debe utilizar el sistema.

- Capacitación por subgrupos de usuarios

Esta forma se utiliza cuando existen diferentes actores que deben utilizar el sistema, por lo tanto, existen diferentes responsabilidades por parte de los usuarios que se hacen cargo de algunas funcionalidades del sistema.

- No hay capacitación

El usuario responsable decide que no se requiere la capacitación.

Migración de datos

La migración de datos se incluye cuando existe una base de datos con información que puede ser utilizada en el nuevo sistema.

Cuando se requiere hacer programas que ayuden a la migración de datos se debe de utilizar este mismo proceso de desarrollo.

Tipos de migración

➤ Migración de datos desde un manejador de bases de datos

Este tipo de migración ocurre cuando se desean aprovechar datos que se encuentran en otras bases de datos y con el manejador de base de datos es posible hacer la exportación. Después con el mismo manejador es posible hacer la importación a la base de datos que utilizará el nuevo sistema. Se deben hacer pruebas de integridad de base de datos para probar que la exportación es correcta.

➤ Migración de conversión de formato

Este tipo de migración ocurre cuando existen datos con diferente formato al que se va utilizar en el nuevo sistema. Es necesario desarrollar los programas para la exportación e importación de datos en base al PDSIIB.

➤ Migración de datos desde archivo

Este tipo de migración ocurre cuando se desea aprovechar datos que se encuentran en archivos de texto. Es necesario desarrollar los programas para la exportación e importación de datos en base al PDSIIB.

TESIS CON
FALLA DE ORIGEN

Estándar para los manuales del sistema

El propósito de este artefacto es indicar las especificaciones que debe tener el manual de configuración, el manual de instalación y el manual de usuario.

Formato:

- ✓ Usar microsoft word 97 o superior.
- ✓ Tipo de letra "Times New Roman" de tamaño 10 para encabezado y pie de pagina, tamaño 12 para texto normal, de tamaño 12 negrita para subtítulos y de tamaño 14 negrita mayúscula para título principal.
- ✓ Dejar dos espacios después de un subtítulo.
- ✓ Dejar un espacio para la separación entre párrafos.
- ✓ Interlineado sencillo.
- ✓ Tabuladores a 1.25 cm.
- ✓ Márgenes superior e inferior a 2.5 cm del borde.
- ✓ Márgenes izquierdo y derecho a 3 cm del borde.
- ✓ Pie de página del .5 cm del borde.
- ✓ El pie de página contendrá el número de página en el centro del documento bajo una línea horizontal.
- ✓ Todo el documento deberá estar justificado.
- ✓ Todas las palabras mayúsculas deberán acentuarse.
- ✓ Impresión a tinta negra en hojas blancas tamaño carta.
- ✓ Espacio entre párrafos de 1 interlineado.
- ✓ El formato para las fechas será el siguiente DD/MM/AAAA donde DD es el día, MM el mes correspondiente y AAAA el año.
- ✓ Las viñetas que se emplearan son las circulares rellenas.
- ✓ Los números que se utilicen deberán estar seguidos de un punto y un espacio.
- ✓ La cabecera del documento debe contener la siguiente leyenda: Nombre del Instituto, nombre del departamento, nombre de la sección, título del manual, (todos estos puntos centrados). En la parte superior derecha deberá ir el escudo del IIB.

Reglas

- ✓ Apoyar las explicaciones con ejemplos.
- ✓ Estructurar en secciones para describir las secuencias de instrucciones con pasos específicos.
- ✓ Incluir imágenes de las pantallas de configuración, instalación o funcionalidad del sistema o de las herramientas que acompañan al sistema.
- ✓ Verificar que las instrucciones sean correctas. Para ello se debe seguir la secuencia de instrucciones de cada explicación y comprobar que no hay inconsistencias.
- ✓ Elaborar en cada ciclo la parte correspondiente al manual evitando que se acumule el trabajo al final del proyecto.
- ✓ Crear un índice y numerar las páginas.

PDSIIB <Artefacto> Plan de distribución

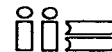
El propósito de este artefacto es indicar los recursos necesarios para poder hacer la entrega a los usuarios finales.

En este artefacto se registran las acciones que debe hacer el usuario responsable para que el sistema se pueda instalar en producción, las herramientas que acompañan al sistema, la forma en que se hará la migración de datos, los manuales que se liberan, la forma y el lugar para realizar la capacitación.

Este artefacto se muestra en la siguiente página.

TESIS CON
FALLA DE ORIGEN

TESIS CON
FALLA DE ORIGEN



Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software

Plan de distribución

Siglas del sistema _____ Ciclo _____
Responsable _____ Fecha _____

Paso: Definir acciones que debe hacer el usuario responsable

Recursos necesarios

Hardware:

Software: _____

Red: _____

Otros: _____

Paso: Definir herramientas que acompañan al sistema

Sistema operativo: _____

Manejador de base de datos: _____

Otros: _____

Paso: Definir recursos materiales para la distribución

# Discos compactos		# Discos flexibles	
# Cartuchos para impresora de inyección de tinta		# Toner para impresora laser	
# Paquetes de hojas blancas			
Otros (especificar)			

Paso: Definir migración de datos

* Usar un manejador de base de datos

Nombre: _____ Versión: _____

* Conversión de formato

Nombre del programa: _____

* Desde archivos

Nombre del programa: _____

* Otro: _____

Paso: Listar casos de uso a entregar

Nombre del módulo o caso de uso	Nombre del módulo o caso de uso

Paso: Listar documentación a liberar

Documentación a liberar	Si	No
Manual de configuración	<input type="checkbox"/>	<input type="checkbox"/>
Manual de instalación	<input type="checkbox"/>	<input type="checkbox"/>
Manual de usuario	<input type="checkbox"/>	<input type="checkbox"/>
Manual técnico	<input type="checkbox"/>	<input type="checkbox"/>
Otro (especificar)	<input type="checkbox"/>	<input type="checkbox"/>

TESTE CON
FALLA DE ORIGEN

**TESIS CON
FALLA DE ORIGEN**

Definir forma de capacitar al usuario (Seleccionar solo una, poner una 'X')

- Capacitación a un usuario responsable
- Capacitación a todos los usuarios
- Capacitación por subgrupos de usuarios
- No hay capacitación

Donde será la capacitación (Seleccionar solo una, poner una "X")

- Auditorio del IIB
- Departamento de Informática
- Sala de usos múltiples del segundo piso
- Sala de usos múltiples del cuarto piso
- Otro (Especificar)

PDSIIB <Artefacto> Invitación de pruebas formales

Este artefacto tiene el propósito de notificar a los usuarios que se llevaran a cabo las pruebas formales de aceptación del sistema en el sitio de desarrollo o en el sitio de producción.

En este artefacto se registra la fecha, la hora, el nombre del sistema, el lugar donde serán las pruebas, casos de uso o módulos a probar y los nombres de los usuarios invitados.

Este artefacto se muestra en la siguiente página.

TESIS CON
FALLA DE ORIGEN

Instituto de Investigaciones Bibliográficas
Departamento de Informática
Sección de desarrollo de Software

Invitación de pruebas formales

Siglas del sistema _____ Ciclo _____
Responsable _____ Fecha _____

Paso: Preparar invitación a los interesados de las pruebas.

Nombre del usuario responsable _____

Por este conducto informamos que el día [Fecha] a las [Hora] se realizarán las pruebas de liberación del sistema [nombre del sistema] en [Lugar donde serán las pruebas] sobre los siguientes módulos.

Lista de casos de uso o módulos

Nombre del caso de uso o módulo	Nombre del caso de uso o módulo

Nombres de los usuarios invitados

Atentamente

Nombre
Administrador de distribución

PDSIIB <Artefacto> Aceptación de pruebas en desarrollo

Este artefacto tiene el propósito de establecer formalmente que las pruebas de aceptación tuvieron éxito y que los usuarios están satisfechos con el sistema.

Este artefacto indica el nombre del sistema y la lista de casos de uso o módulos cuando es una entrega parcial.

La aceptación de pruebas en desarrollo la firma el usuario responsable del sistema y, con ello, se da por entendido que los usuarios aceptan que la funcionalidad del sistema es correcta.

Este artefacto se muestra en la siguiente página.

TESIS CON
FALLA DE ORIGEN

TESIS CON
FALLA DE ORIGEN



Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software

Aceptación de pruebas en desarrollo

Siglas del sistema _____ Ciclo _____
Responsable _____ Fecha _____

Por este conducto informamos que el sistema [nombre del sistema] cumple con los requerimientos establecidos en los siguientes casos de uso:

Lista de casos de uso o módulos liberados

Nombre del caso de uso o módulo	Nombre del caso de uso o módulo

Atentamente

Nombre
Administrador de distribución

Vo. Bo.

Nombre
Usuario responsable

PDSIIB <Artefacto> Informe de entregable

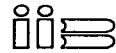
Este artefacto tiene el propósito de listar las herramientas que acompañan al sistema y listar los manuales que se van a entregar al administrador del proyecto una vez que se ha validado que el sistema cumple con los requerimientos de aceptación

Este informe incluye las "Especificaciones del sistema en internet" cuando el sistema se encuentra instalado en el servidor web del Departamento de Informática y no es necesario entregarlo en discos. En éstas especificaciones se indica el nombre del sistema, la dirección electrónica donde se encuentra el sistema listo para su funcionamiento, la cuenta de usuario, la contraseña a utilizar, el nombre del visualizador web que se debe usar y la versión del mismo.

Este artefacto se muestra en la siguiente página.

TESIS CON
FALTA DE ORIGEN

TESIS CON
FALLA DE ORIGEN



Instituto de Investigaciones Bibliográficas
Departamento de Informática
Sección de desarrollo de Software

Informe de entregable

Siglas del sistema _____ Ciclo _____
Responsable _____ Fecha _____

Nombre del sistema a entregar _____

Paso: Listar manuales a entregar

Seleccionar si se hace la entrega	Si	No
Manual de configuración	<input type="checkbox"/>	<input type="checkbox"/>
Manual de instalación	<input type="checkbox"/>	<input type="checkbox"/>
Manual de usuario	<input type="checkbox"/>	<input type="checkbox"/>
Manual técnico	<input type="checkbox"/>	<input type="checkbox"/>

Paso: Listar herramientas que acompañan al sistema

Sistema operativo: _____

Manejador de base de datos: _____

Otros: _____

Paso: Realizar especificaciones del sistema en internet

Nombre del sistema _____

Dirección electrónica http:// _____

ftp:// _____

Cuenta _____

Contraseña _____

Visualizador que debe utilizar _____ Versión _____

Atentamente

Nombre
Administrador de distribución

Vo. Bo.

Nombre
Administrador del proyecto

TESIS CON
FALLA DE ORIGEN

**TESIS CON
FALLA DE ORIGEN**

PDSIIB <Artefacto> Aceptación de pruebas en sitio

Este artefacto tiene el propósito de establecer formalmente que las pruebas de aceptación en sitio tuvieron éxito y que los usuarios están satisfechos con el sistema.

Este artefacto indica el nombre del sistema liberado y la lista de casos de uso o módulos cuando es una entrega parcial.

La aceptación de pruebas en sitio se entrega al administrador del proyecto.

Este artefacto se muestra en la siguiente página.

Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software



Aceptación de pruebas en sitio

Siglas del sistema _____ Ciclo _____
Responsable _____ Fecha _____

Por este conducto informamos que el sistema [nombre del sistema] cumple con los requerimientos establecidos en los siguientes casos de uso:

Lista de casos de uso o módulos liberados

Nombre del caso de uso o módulo	Nombre del caso de uso o módulo

Vo. Bo.

Nombre
Usuario responsable

Recibió

Nombre
Administrador del proyecto

TESIS CON
FALLA DE ORIGEN

TESIS CON
FALLA DE ORIGEN

Capítulo 5

Flujo de trabajo pruebas

El propósito de este flujo es asegurar que el sistema cumple con los requerimientos establecidos. La descripción a detalle de acuerdo al Proceso Unificado de Rational se encuentra en el Apéndice C.

En este capítulo se explica como es el flujo de trabajo pruebas, las consideraciones para hacer la adaptación y el resultado de la adaptación de cada actividad y sus pasos. Se hace una explicación de los artefactos y para los roles se hace una descripción y se describe el perfil.

Para explicar cada actividad se tiene el siguiente formato: se pone del lado izquierdo la etiqueta RUP, el nombre de la actividad y el nombre del rol responsable de la actividad, de acuerdo a como lo establece RUP. Después se pone la etiqueta ADAPTACION y el nombre de la actividad y enseguida las consideraciones para hacer la adaptación. Después se utiliza la etiqueta PDSIIB para designar al proceso de desarrollo de sistemas del Instituto de Investigaciones Bibliográficas en el Departamento de Informática, se pone el nombre de la actividad como quedó en el proceso resultante y el nombre del rol responsable. Cuando finaliza la descripción de la actividad se pone a nivel de pasos cada actividad con la etiqueta PDSIIB y el nombre del paso con su descripción.

Actividades

RUP <Actividad> Hacer el plan de pruebas
<Rol> Diseñador de pruebas

El propósito de esta actividad es coleccionar y organizar la información para planear las pruebas y crear el plan de pruebas.

Los requerimientos para las pruebas identifican lo que será probado junto con el alcance y el propósito del esfuerzo de las pruebas.

Las fuentes más comunes de requerimientos incluye la lista de requerimientos, los casos de uso, el modelo de casos de uso, las realizaciones de casos de uso, especificaciones suplementarias, entrevistas y la revisión de sistemas existentes.

Se indica que requerimientos van a ser probados. Esto conduce a jerarquizar los requerimientos según su importancia.

Se hace la identificación de riesgos para hacer las pruebas, los requerimientos más importantes para las pruebas son aquéllos que reflejan los riesgos más altos.

Cada requerimiento a probar debe revisarse e identificar el factor de riesgo como alto, medio o bajo.

Se hace el desarrollo de las estrategias de las pruebas y se comunica al equipo de desarrollo.

TESIS CON
FALLA DE ORIGEN

TESIS CON
FALLA DE ORIGEN

**TESIS CON
FALLA DE ORIGEN**

Consideraciones para establecer las estrategias:

- Identificar y describir como se probará el desarrollo.
- Identificar el criterio para hacer las pruebas
- Identificar cualquier consideración especial

Se describe cómo se llevarán a cabo las pruebas. Las acciones a tomar mientras se está probando y cómo se verificarán los resultados

Se identifican los recursos para hacer las pruebas, quién hará las pruebas y lo que se necesita para apoyar estas actividades, identificando los recursos necesarios.

Se hace el calendario para las pruebas considerando, habilidades de la persona que realiza la prueba, nivel de conocimiento de los recursos humanos que trabajan en el proyecto, los parámetros sobre la aplicación (número de ventanas, componentes, entidades de datos, relaciones y porcentaje de reuso) y la cobertura de las pruebas donde muchos casos de prueba se requieren para la aceptación de los casos de uso o requerimientos

ADAPTACIÓN <Actividad> Hacer el plan de pruebas

Las pruebas de los sistemas están basadas en los casos de uso debido a que éstos proporcionan una buena forma de medir el avance a la hora de ejecutar las pruebas.

Las estrategias de prueba a utilizar para el proceso se listan en el estándar de pruebas, tales como pruebas de unidad, de función, de ciclo de negocio, de base de datos, de interfaz de usuario, de instalación y de configuración. El estándar de pruebas indica los niveles de pruebas y las estrategias de pruebas que serán utilizadas en el Departamento de Informática.

PDSIIB <Actividad> Hacer el plan de pruebas
<Rol> Diseñador de pruebas

En el plan de pruebas se listan los casos de uso que serán probados, se asigna el nombre a los casos de prueba y se identifican los recursos para hacer las pruebas.

RUP <Actividad> Diseñar las pruebas
<Rol> Diseñador de pruebas

El propósito de esta actividad es identificar un conjunto de casos de prueba comprobables para cada construcción e identificar los procedimientos de prueba.

Se deben describir acciones y los pasos del actor al usar el sistema. Las descripciones de los procedimientos de prueba son utilizadas para identificar y describir los casos de prueba necesarios para probar la aplicación.

Las entradas primarias para identificar casos de prueba son: los casos de uso que en algún punto recorren algún destino de las pruebas como el sistema, subsistema o componente, el modelo de diseño, cualquier requerimiento técnico o suplementario, mapa de la aplicación del objetivo de las pruebas.

Para describir los casos de prueba se debe indicar: las condiciones de la prueba, el caso de uso, escenario del caso de uso o los requerimientos técnicos de casos de prueba derivados, el resultado esperado y condición o valores de datos.

Se debe identificar y estructurar los procedimientos de prueba revisando el flujo de la aplicación, desarrollando el modelo de pruebas y estructurando los procedimientos de prueba.

Se deben identificar medidas para la cobertura de las pruebas, hay dos métodos para determinar la cobertura de las pruebas: cobertura basada en requerimientos y cobertura basada en el código.

La cobertura basada en requerimientos utiliza casos de uso, requerimientos, flujos de casos de uso o condiciones de prueba como la medida del total de elementos probados, además de que se usarán para hacer el plan de pruebas.

La cobertura basada en el código utiliza el código generado como el elemento para conocer si las pruebas están completas y como medida de que el código ha sido ejecutado durante las pruebas (como las líneas de código ejecutadas o el número de caminos recorridos). Este tipo de medida sólo puede llevarse a cabo después de que el código ha sido generado.

Identificar el método a utilizar y mencionar cómo se realizará la medición, cómo deben interpretarse los datos y cómo se usará la métrica durante el proceso.

ADAPTACIÓN <Actividad> Diseñar las pruebas

El diseño de las pruebas se hace en base al estándar de pruebas y al plan de pruebas, una vez seleccionadas las estrategias de prueba, se hace la especificación del caso de prueba, para ello, se utiliza el artefacto "caso de prueba" donde se indican las precondiciones, las postcondiciones y el procedimiento de prueba. En el procedimiento de prueba se indica la secuencia de acciones a realizar, la entrada de datos y el resultado esperado.

La cobertura de las pruebas será medida en función de los casos de uso y los casos de prueba ejecutados para cada uno de ellos, por lo tanto, se completará la cobertura cuando hayan sido ejecutados todos los casos de prueba de los casos de uso, es decir, la cobertura se hace en base a los requerimientos.

PDSIIB <Actividad> Hacer el diseño de las pruebas

<Rol> Escritor técnico

Consiste en describir en detalle los casos de prueba que se listaron en el plan de pruebas.

RUP <Actividad> Ejecutar las pruebas

<Rol> Probador

El propósito de esta actividad es ejecutar las pruebas y registrar los resultados.

Se debe configurar el ambiente de pruebas para asegurar que todos los componentes necesarios se han implementado y se encuentran en el ambiente.

TESIS CON
FALLA DE ORIGEN

Los resultados de las pruebas deben revisarse para asegurar que son fiables e informar de los fracasos, advertencias o resultados inesperados no causados por influencias externas, como puede ser una configuración inadecuada.

ADAPTACIÓN <Actividad> Ejecutar las pruebas

Los resultados de las pruebas serán registrados en el artefacto "caso de prueba" en la casilla de resultado de la prueba y además se debe anotar la fecha de ejecución de la prueba.

PDSIIB <Actividad> Ejecutar pruebas
<Rol> Probador

Consiste en ejecutar el procedimiento de los casos de prueba y registrar los resultados. Esta actividad tiene exactamente los mismos pasos para los diferentes niveles de pruebas que son: pruebas unitarias, pruebas de integración y pruebas de sistema.

Para las pruebas de aceptación en desarrollo y pruebas de aceptación in sitio que están ubicados en el nivel de pruebas de sistema también se utilizan los pasos de ésta actividad.

RUP <Actividad> Evaluar las pruebas
<Rol> Diseñador de pruebas

El propósito de está actividad es evaluar los resultados de las pruebas y hacer solicitudes de cambio, calcular y entregar medidas de las pruebas y generar el reporte de evaluación.

Se deben analizar los resultados de las pruebas para identificar detalles con respecto a diferencias entre los resultados obtenidos y los resultados esperados. Las diferencias indican defectos potenciales en las pruebas y deben registrarse en la solicitud de cambios para tomar las acciones correctivas.

Para evaluar la cobertura de las pruebas basadas en requerimientos es necesario revisar los resultados de las pruebas para determinar: la proporción entre cuántas pruebas basadas en requerimientos (casos de prueba) han sido ejecutados en esta iteración y el número total de pruebas. También se debe estimar la proporción de casos de prueba con éxito.

Para evaluar la cobertura de las pruebas basado en el código es necesario revisar los resultados y determinar: la proporción entre el código que se ha ejecutado durante las pruebas en esta iteración y el código total del objetivo de las pruebas.

Finalmente se debe generar el resumen de evaluación de prueba.

ADAPTACIÓN <Actividad> Evaluar las pruebas

La evaluación de las pruebas se establece a través de una cobertura basada en requerimientos. Para ello, se deben registrar el número casos de prueba con error y el número de casos de prueba sin error.

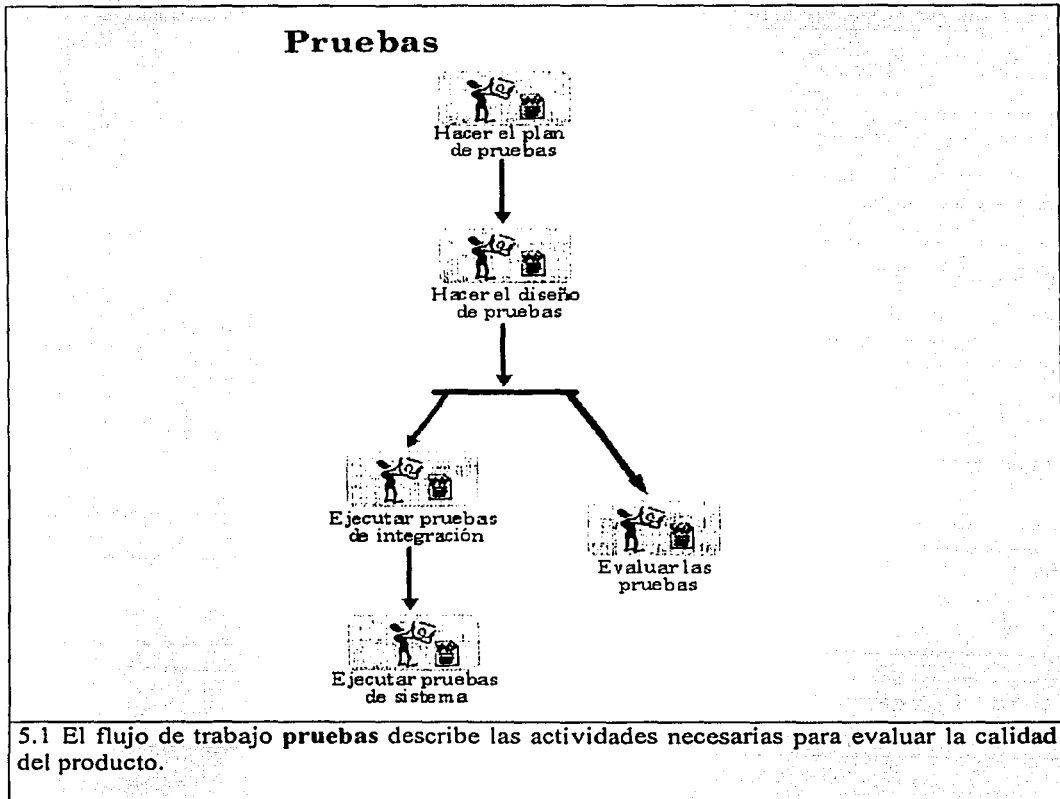
La proporción de errores se obtiene con la formula siguiente:

Proporción de error = número de casos con error : número de casos con error.

PDSIIB <Actividad> Evaluar las pruebas
<Rol> Diseñador de pruebas

Consiste en estimar la calidad del sistema, hacer solicitudes de cambio y conocer la cobertura de las pruebas.

El nuevo flujo de trabajo pruebas se puede apreciar en el diagrama 5.1.



PDSIIB <Actividad> Hacer el plan de pruebas
<Rol> Diseñador de pruebas

PDSIIB <Paso> Reunir artefactos de entrada

Reunir los artefactos de entrada necesarios correspondientes a esta actividad. Los artefactos son: Modelo de análisis y estándar de pruebas.

PDSIIB <Paso> Identificar casos de uso a probar

Consiste en registrar en el plan de pruebas los casos de uso a probar, los nombres de los casos de uso se obtienen del modelo de análisis. Se registran los nombres de los casos de uso o módulo a probar, se asigna el nombre al caso de prueba. Los casos de uso a probar se obtienen del artefacto "Modelo de análisis".

Consultar el estándar de pruebas.

PDSIIB <Paso> Identificar recursos para hacer las pruebas

Consiste en registrar en el plan de pruebas quien será el responsable de hacer las pruebas, el hardware y software necesario y la configuración de la red.

El responsable para hacer las pruebas debe tener la capacidad para planear pruebas, diseñarlas, ejecutarlas y analizar los resultados.

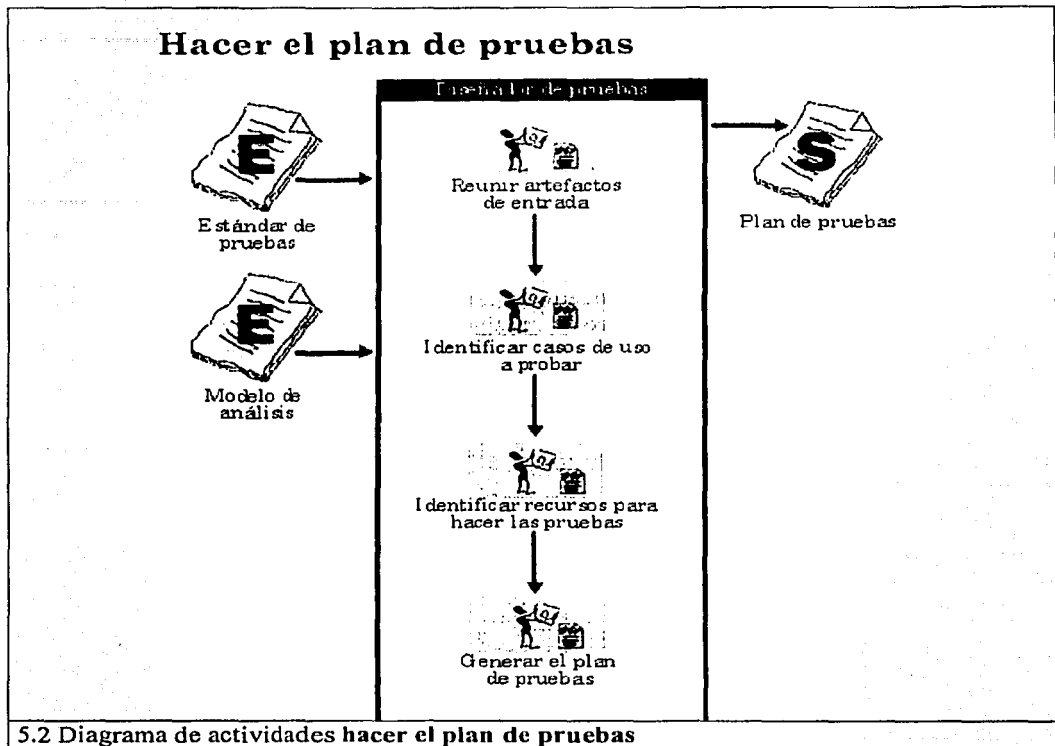
PDSIIB <Paso> Generar el plan de pruebas

Consiste en revisar la información del plan de pruebas para asegurar que contiene los casos de uso o módulos a probar y los recursos para hacer las pruebas.

Artefactos de salida

Plan de pruebas.

La actividad **hacer el plan de pruebas** se puede apreciar en el diagrama 5.2.



PDSIIB <Actividad> Hacer el diseño de pruebas
<Rol> Diseñador de pruebas

PDSIIB <Paso> Reunir artefactos de entrada

Reunir los artefactos de entrada necesarios correspondientes a esta actividad. Los artefactos son: Plan de pruebas, estándar de pruebas y modelo de implementación.

PDSIIB <Paso> Identificar los casos de prueba

Consiste en revisar el plan de pruebas para obtener la lista completa de los casos de prueba que se van a describir en el artefacto "Caso de prueba".

PDSIIB <Paso> Describir los casos de prueba

Consiste en registrar para cada "caso de prueba" el nombre del caso de uso a probar, el nombre del caso de prueba, las precondiciones, las postcondiciones y el procedimiento de prueba.

En el procedimiento de prueba del "Caso de prueba" se registran el número de cada paso, la acción a realizar, las entradas de datos y el resultado esperado.

PDSIIB <Paso> Verificar el procedimiento en cada caso de prueba

Consiste en verificar que el procedimiento de prueba es correcto de acuerdo a la secuencia de acciones a realizar, datos de entrada y datos del resultado esperado.

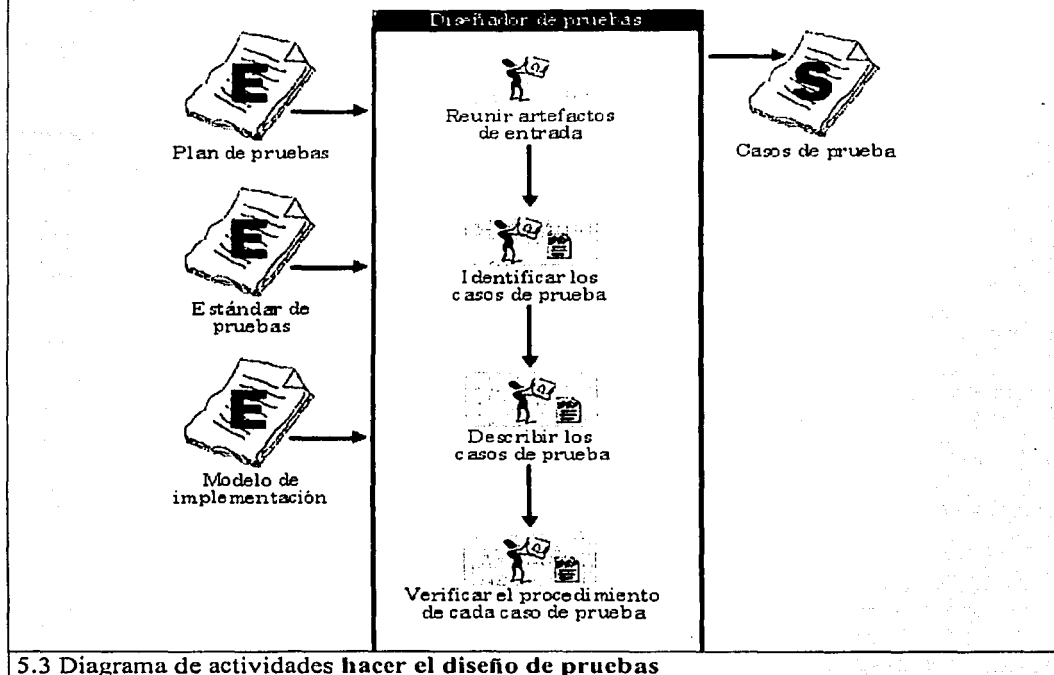
Muchos casos de prueba son variantes de otros, lo que significa que pueden ser reutilizados por contener acciones comunes. Se debe guardar el caso de prueba con otro nombre y cambiar únicamente las especificaciones que hacen un nuevo caso de prueba.

Artefactos de salida

Varios casos de prueba.

La actividad **hacer el diseño de pruebas** se puede apreciar en el diagrama 5.3.

Hacer el diseño de pruebas



5.3 Diagrama de actividades hacer el diseño de pruebas

PDSIIB <Actividad> Ejecutar pruebas
<Rol> Probador

PDSIIB <Paso> Reunir artefactos de entrada

Reunir los artefactos de entrada necesarios correspondientes a esta actividad. Los artefactos son: Componentes, subsistema, sistema, casos de prueba, estándar de pruebas y plan de pruebas.

PDSIIB <Paso> Ejecutar los casos de prueba

Consiste en llevar a cabo el procedimiento de prueba definido a través de las secuencias de acciones del caso de prueba y registrar el resultado.

Configurar el ambiente de pruebas para asegurar que todos los componentes se encuentran en el ambiente.

**TESIS CON
FALLA DE ORIGEN**

Iniciar el ambiente de pruebas y revisar que la aplicación se encuentra en el estado inicial correcto para comenzar con las pruebas.

PDSIIB <Paso> Verificar los resultados de las pruebas

Consiste en revisar que los resultados son fiables y en asegurar que se ha registrado el resultado de la prueba.

Revisar que los resultados inesperados no son causa de influencias externas como puede ser una configuración inadecuada, falta de componentes, falta de archivos, etc.

PDSIIB <Paso> Recuperar el sistema de pruebas fallidas

Consiste en restaurar el sistema para continuar con las pruebas.

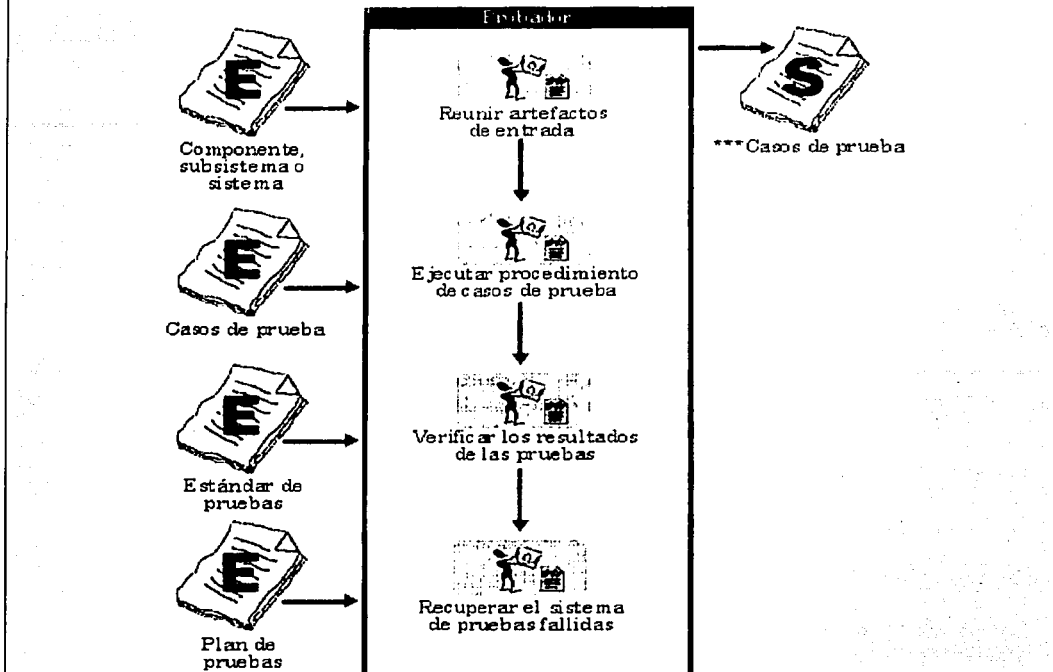
Determinar la causa del problema y corregirlo, reconfigurar el ambiente si es necesario, reiniciar el ambiente de pruebas y volver a ejecutar las pruebas.

Artefactos de salida

Varios casos de prueba.

La actividad **ejecutar pruebas** se puede apreciar en el diagrama 5.4.

Ejecutar pruebas



5.4 Diagrama de actividades ejecutar pruebas

PDSIIB <Actividad> Evaluar las pruebas
<Rol> Diseñador de pruebas

PDSIIB <Paso> Reunir artefactos de entrada

Reunir los artefactos de entrada necesarios correspondientes a esta actividad. Los artefactos son: Componentes, subsistema, sistema, casos de prueba y estándar de pruebas.

TESIS CON
 FALLA DE ORIGEN



PDSIIB <Paso> Hacer solicitudes de cambio

Consiste en indicar las causas del problema para que sean corregidas.

Revisar los resultados de la prueba para identificar causas con respecto a los resultados esperados y los resultados de la prueba. Las diferencias muestran defectos potenciales que deben indicarse para tomar las acciones correctivas.

PDSIIB <Paso> Evaluar la cobertura de las pruebas

Consiste en contar el número de casos de prueba ejecutados, el número de caso de prueba sin error y el número de casos de prueba con error y registrarlos en el artefacto "Reporte de evaluación de pruebas".

La cobertura de las pruebas se mide por el número de casos de prueba ejecutados.

PDSIIB <Paso> Estimar la proporción de defectos

Consiste en registrar en el artefacto "Reporte de evaluación de pruebas" el número de casos de prueba con error y el número de casos de prueba sin error.

También se deben registrar los nombres de los casos de prueba que tuvieron error.

PDSIIB <Paso> Entregar las solicitudes de cambio

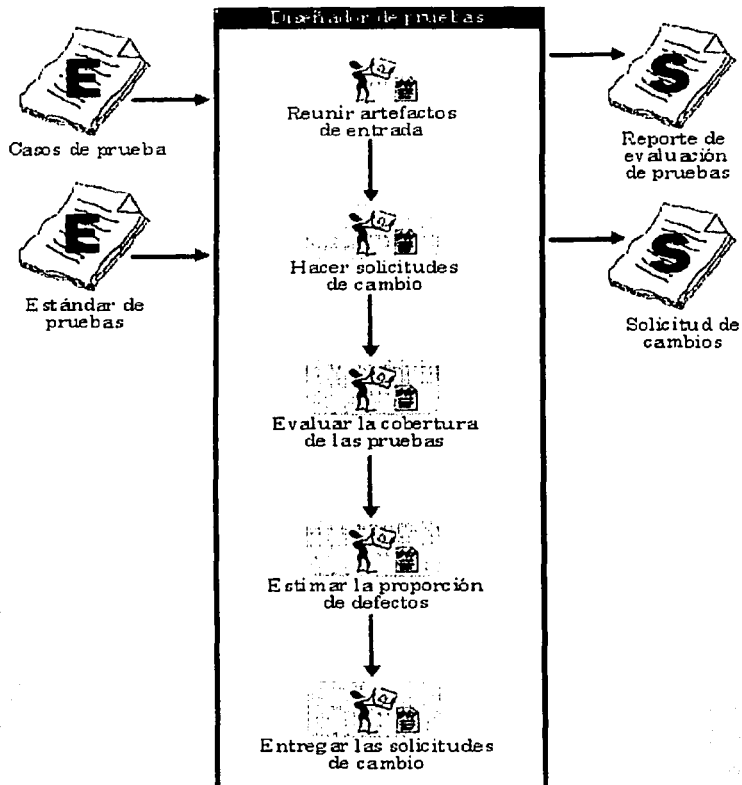
Consiste en entregar las solicitudes de cambio al administrador de cambios.

Artefactos de salida

Solicitudes de cambio y el reporte de evaluación de pruebas.

La actividad **evaluar las pruebas** se puede apreciar en el diagrama 5.5.

Evaluar las pruebas



5.5 Diagrama de actividades evaluar las pruebas

TESIS CON
FALLA DE ORIGEN

Roles

PDSIIB <Rol> Diseñador de pruebas

Es el encargado de hacer la planeación, el diseño y la implementación de las pruebas. Esto involucra la generación del plan de pruebas y la implementación de los casos de prueba.

Perfil del rol

Tener conocimiento en ambientes de desarrollo de sistemas y la habilidad para identificar requerimientos para probar.

PDSIIB <Rol> Probador

Es el encargado de ejecutar las pruebas y registrar los resultados en los casos de prueba.

Perfil del rol

Tener conocimiento en ambientes de desarrollo de sistemas y la habilidad para seguir los pasos que comprende un procedimiento de prueba.

Artefactos

PDSIIB <Artefacto> Estándar de pruebas

En este artefacto se describen los niveles y estrategias de prueba, como hacer el diseño de las pruebas, como hacer la verificación de los resultados de las pruebas y como se hace la evaluación.

Este estándar se debe utilizar como ayuda para realizar los casos de prueba, ya que indica las recomendaciones de acuerdo a la estrategia de prueba.

Este artefacto se muestra en la siguiente página.



Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software



Estándar de pruebas

El propósito de este documento es describir los niveles y estrategias de pruebas a realizar durante el proceso de desarrollo.

Niveles de pruebas

➤ Pruebas unitarias

Son el conjunto de acciones que se realizan para probar los elementos más pequeños de un programa como son: casos de uso, métodos, procedimientos y funciones.

En este nivel de pruebas se utilizan las estrategias de prueba de unidad (caja blanca) y pruebas de función (caja negra).

➤ Pruebas de integración

Son el conjunto de acciones que se realizan para probar instrucciones después de integrar objetos, casos de uso, módulos, funciones o procedimientos y que cumplan con el comportamiento esperado.

En este nivel de pruebas se utilizan las estrategias de prueba de función (caja negra), pruebas de ciclo de negocio y pruebas de interfaz de usuario.

➤ Pruebas de sistema

Son el conjunto de acciones que se realizan para probar que el sistema cumple con las funcionalidades de acuerdo a los requerimientos del usuario. En este tipo de pruebas el usuario no está presente.

En este nivel de pruebas se utilizan las estrategias de prueba de función (caja negra), pruebas de ciclo de negocio, pruebas de interfaz de usuario, pruebas de instalación, pruebas de configuración y pruebas de base de datos.

➤ Pruebas de aceptación en desarrollo

Son pruebas de sistema con la variante de que interviene el usuario, las pruebas se hacen en el Departamento de Informática, lugar donde normalmente se desarrollan los sistemas.

En este nivel de pruebas se utilizan las estrategias de prueba de función (caja negra) y pruebas de ciclo de negocio principalmente. Se pueden utilizar las estrategias de pruebas de interfaz de usuario, pruebas de instalación y pruebas de configuración si son necesarias.

➤ Pruebas de aceptación en sitio

Son pruebas de sistema con la variante de que se hacen en el lugar donde el usuario va a utilizar el sistema, es decir, en producción. El usuario debe estar presente para validar la liberación.

En este nivel de pruebas se utilizan las estrategias de prueba de función (caja negra) y pruebas de ciclo de negocio principalmente. Se pueden utilizar las estrategias de pruebas de interfaz de usuario, pruebas de instalación y pruebas de configuración si son necesarias.

Estrategias de prueba

➤ Pruebas de unidad (caja blanca)

Se enfocan a probar diferentes caminos a través de las estructuras de control de código.

Recomendaciones: probar ejecutando por lo menos una vez cada instrucción del programa, ejecutar cada condición del programa que tenga operadores como \neq , \leq , \geq , $<$, $=$, $>$, or, and, not, etc. Ejecutar el flujo de datos para las variables, ejecutar una vez un ciclo, ejecutar dos veces un ciclo, ejecutar el ciclo en diferentes rangos de valores.

➤ Pruebas de función (caja negra)

Se enfocan a probar los requerimientos proyectados en los casos de uso. El objetivo de este tipo de pruebas es verificar la introducción de datos, el procesamiento y recuperación de los mismos. Con ésta estrategia se verifica que el sistema obtiene los resultados esperados.

Recomendaciones: probar límites inferiores, límites superiores, valores de datos válidos, valores de datos no válidos, Límites inferiores no validos, límites superiores no válidos, formatos válidos, formatos no válidos, etc.

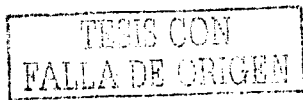
➤ Pruebas de ciclo de negocio

Se enfocan a simular acciones del usuario sobre el sistema en ciertos periodos de tiempo en las que se hacen ciertas funcionalidades o transacciones.

Recomendaciones: Probar periodos especiales válidos, periodos especiales no validos, fechas especiales válidas, fechas especiales no válidas, horas válidas, horas no válidas, etc.

➤ Pruebas de base de datos

Se enfocan a revisar que las transacciones que se hacen con la base de datos son correctas, que los datos almacenados cumplen con las especificaciones de validación y formato, que las relaciones entre llaves primarias y foráneas sean las adecuadas.



Las pruebas a la base de datos se llevan a cabo de dos maneras:

1. Base de datos en bruto: Está enfocada a probar la base de datos sin intervención de ningún caso de uso o módulo del sistema.

Se deben hacer varios casos de prueba para la base de datos.

Recomendaciones: Probar llaves primarias, llaves foráneas, actualización en cascada, borrado en cascada, eventos válidos, eventos no válidos, ejecución de procedimientos almacenados, datos válidos, datos no válidos, formatos válidos, formatos no válidos, datos de retorno, generación de índices, etc.

2. Base de datos con interacción del sistema: Está enfocada a probar que la base de datos manifiesta las transacciones realizadas a través del sistema.

Recomendaciones: Probar que el sistema agregó un registro en la base de datos, que se borró un registro, que fue actualizado el registro, que se actualizaron los registros que están relacionados, que se borraron los registros que están relacionados, que se registraron los datos con el formato deseado a través del sistema, etc.

➤ Pruebas de interfaz de usuario

Se enfocan a probar la interacción que tendrá el usuario con el sistema, el objetivo es asegurar que la interfaz cumple con los requerimientos apropiados para acceder y navegar a través del sistema.

Recomendaciones: probar con ventana inicial, ventana siguiente, ventana anterior, ventana de error, ventana de mensaje, campo siguiente, campo anterior, tabulador, tamaño de ventana, posición de la ventana, estado de la ventana, prueba de despliegue de menús, enfoque de objetos, etc.

➤ Pruebas de instalación

Se enfoca a verificar que el software se instala bajo diferentes condiciones como una nueva instalación, una actualización o una instalación completa y que opere correctamente.

Recomendaciones: probar que opere el sistema después de instalar, hacer una nueva instalación, hacer una nueva instalación con la misma versión, hacer una instalación con una versión anterior, etc.

➤ Pruebas de configuración

Se enfocan a verificar que el sistema funcione en diferentes tipos de equipo de cómputo, que funcione en donde están instalados otros sistemas o aplicaciones, que funcione en diferentes plataformas de hardware y software.

Recomendaciones: Probar con otras aplicaciones instaladas junto con el sistema, diferentes sistemas operativos, diferentes versiones de sistema operativo, diferente plataforma, etc.

Utilizar la siguiente tabla para nombrar los casos de prueba

Estrategia de prueba	Prefijo para nombrar los casos de prueba
Pruebas de unidad (caja blanca)	PU
Pruebas de función (caja negra)	PF
Pruebas de ciclo de negocio	PCN
Pruebas de base de datos	PBD
Pruebas de base de datos en bruto	PBDB
Pruebas de interfaz de usuario	PIU
Pruebas de instalación	PI
Pruebas de configuración	PC

Conjunto de datos de prueba

El usuario responsable tiene la obligación de entregar datos de prueba, estos datos los puede entregar en papel, en archivo o en una base de datos.

Diseño de pruebas

El diseño de las pruebas se hace a través de la especificación de precondiciones, postcondiciones y del procedimiento de prueba, definición de datos de entrada y el resultado esperado en el artefacto "Caso de prueba".

Verificación de resultados de las pruebas

La verificación se hace cuando al ejecutar el procedimiento de los casos de prueba el resultado esperado es diferente al resultado de la prueba. Se debe revisar la configuración del sistema en base al manual de configuración, al manual de instalación y manuales de herramientas que acompañan al sistema.

Evaluación de pruebas

Para hacer la evaluación de las pruebas se toma en cuenta la cobertura de las pruebas y la proporción de defectos encontrados.

➤ Cobertura de pruebas

Indica cuantos casos de prueba se han ejecutado. Cuando se ha realizado el total de casos de prueba la cobertura está completa.

➤ Proporción de defectos

Es una indicación del número de errores que tiene el componente, subsistema o sistema que se está probando con respecto a los casos de prueba que se ejecutaron con éxito.

TESIS CON
FALLA DE ORIGEN

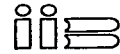
PDSIIB <Artefacto> Plan de pruebas

En este artefacto se listan los casos de uso, los nombres de los casos de prueba y los recursos necesarios para hacer las pruebas
Para llenar el plan de pruebas se debe tener a la mano el estándar de pruebas.

Este artefacto se muestra en la siguiente página.

Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software



Plan de pruebas

Siglas del sistema _____ Ciclo _____
Responsable _____ Fecha _____

Paso: Identificar casos de uso a probar

Utilizar la siguiente tabla para nombrar los casos de prueba

Estrategia de prueba	Prefijo para nombrar los casos de prueba
Pruebas de unidad (caja blanca)	PU_
Pruebas de función (caja negra)	PF_
Pruebas de ciclo de negocio	PCN_
Pruebas de base de datos	PBD_
Pruebas de interfaz de usuario	PIU_
Pruebas de instalación	PI_
Pruebas de configuración	PC_

Caso de uso o módulo	Caso de prueba
	PU_
	PF_
	PCN_
	PBD_
	PIU_
	PU_
	PF_
	PCN_
	PBD_
	PIU_
	PU_
	PF_
	PCN_
	PBD_
	PIU_

Nombre del Sistema	Caso de prueba
	PI_
	PC_

TESIS CON
FALLA DE ORIGEN

**TESIS CON
FALLA DE ORIGEN**

Paso: Identificar recursos para hacer las pruebas

Responsable de hacer las pruebas _____

Hardware _____

Software _____

Red _____

Otros (especificar) _____

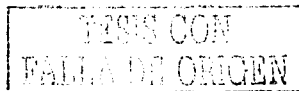
PDSIIB <Artefacto> Caso de prueba

En este artefacto se indica el nombre del caso de uso o módulo a probar, el nombre del caso de prueba, precondiciones, postcondiciones, procedimiento de prueba, entrada de datos, resultado esperado y resultado de la prueba.

La finalidad de este artefacto es actuar como guía para realizar las pruebas y verificar los resultados. Después servirá para saber si se tiene que hacer una solicitud de cambio.

Se debe tener a la mano el estándar de pruebas donde se indican los aspectos pertinentes para probar según la estrategia de prueba.

Este artefacto se muestra en la siguiente página.



TESIS CON
FALLA DE ORIGEN



Instituto de Investigaciones Bibliográficas
Departamento de Informática
Sección de desarrollo de Software

Caso de prueba

Siglas del sistema _____ Ciclo _____
Responsable _____ Fecha _____

Paso: Identificar los casos de prueba

Nombre del caso de uso o módulo a probar _____

Paso: Describir los casos de prueba

Nombre del caso de prueba _____

Precondiciones

Postcondiciones

Procedimiento de prueba

#	Acción a realizar	Entrada de datos	Resultado esperado	Resultado de la prueba

Fecha de ejecución _____

PDSIIB <Artefacto> Reporte de evaluación de pruebas

En este artefacto se indica el número de casos de prueba ejecutados, el número de casos de prueba sin error y el número de casos de prueba con error. También se registra la lista de nombre de los casos de prueba que tuvieron algún error.

La finalidad de este artefacto es conocer la calidad del componente, subsistema o sistema y sirve como base para hacer solicitudes de cambio.

Se debe tener a la mano el estándar de pruebas.

Este artefacto se muestra en la siguiente página.

YES CON
FALLA DE ORIGEN

Instituto de Investigaciones Bibliográficas
Departamento de Informática
Sección de desarrollo de Software

Reporte de evaluación de pruebas

Siglas del sistema _____ Ciclo _____
Responsable _____ Fecha _____

Paso: Evaluar las pruebas

de casos de prueba ejecutados _____

de casos de prueba sin error _____

de casos de prueba con error _____

Proporción de error: # de casos con error : # de casos sin error
_____ : _____

Lista de casos de prueba con error

Lista de casos de prueba con error	

Observaciones

--

Capítulo 6

Herramienta para el PDSIIB

El propósito de la herramienta es guiar el Proceso de Desarrollo de Sistemas del Departamento de Informática del Instituto de Investigaciones Bibliográficas (PDSIIB).

Planteamiento de la herramienta

La herramienta modela el proceso de desarrollo a través de diagramas de actividades que muestran el contexto de cada actividad indicando el rol responsable, los pasos a seguir y los artefactos de entrada y salida que se deben utilizar. También brinda ayuda a personas que desean introducirse en los quehaceres del desarrollo de sistemas.

Para crear la herramienta se utilizó el hipertexto, editores para manejo de imágenes, procesadores de palabras, editores de páginas web y el servidor web.

El hipertexto esta basado en etiquetas del lenguaje de marcado de hipertexto mejor conocido como html, se hicieron ligas a los diagramas de actividades y a los artefactos.

El editor de páginas web se utiliza para hacer las páginas de la herramienta, cuando haya que crear nuevos diagramas de actividad o artefactos a introducir en el proceso definido, se puede hacer fácilmente con ayuda del editor.

Los editores de imágenes se utilizaron para crear los símbolos y diagramas de actividad de los flujos de trabajo.

El procesador de palabras se utilizó para hacer los artefactos.

El servidor web se utilizo para poner a disposición del personal del Departamento de Informática la herramienta.

Se hizo el diseño de la herramienta con el objetivo de que sea extensible, y esto se logra debido al uso del hipertexto, si en algún momento es necesario agregar o quitar diagramas o artefactos simplemente se agregan o se quitan las ligas.

Objetivos

- Crear una herramienta que contenga la descripción de todo el proceso para que sirva como guía en el desarrollo de sistemas del Departamento de Informática.
- Colaborar en la integración de los flujos de trabajo ambiente, distribución y pruebas con los demás flujos adaptados de [Sánchez] y [Ramírez] en una herramienta.

Diagramas de actividades

Estos diagramas representan el flujo de actividades que se deben realizar durante el proceso de desarrollo de un sistema.

Niveles de diagramas:

- Diagramas de fase: Muestran las actividades que tiene una fase.
- Diagramas por actividad: Muestran los pasos que tiene una actividad.
 - En la parte izquierda del diagrama se muestran los artefactos de entrada que serán utilizados al realizar algún paso.
 - En la parte central del diagrama se muestran los pasos a realizar. Los pasos están delimitados en el diagrama por carriles que indican el rol responsable para realizar los pasos.
 - En la parte derecha del diagrama se muestran los artefactos de salida que han sido utilizados en algún paso.
 - Cuando en un artefacto de salida aparecen tres asteriscos ***, significa que el artefacto ya debe tener datos y que se van a agregar otros más.

Uso de la herramienta

Para navegar a través del proceso se debe pulsar sobre el nombre de una fase, actividad, paso o artefacto para obtener una descripción a mayor detalle.

Pulsar sobre el nombre de una fase significa obtener un nuevo diagrama que corresponde a las actividades de una fase.

Pulsar sobre el nombre de un artefacto significa obtener el documento real o una explicación de cómo se utiliza el artefacto.

Pulsar sobre el nombre de una actividad significa obtener el diagrama de actividad mostrando los pasos que corresponden a esa actividad.

Pulsar sobre el nombre de un paso significa obtener la explicación de lo que se debe hacer en ese paso.

Las flechas representan el flujo secuencial entre las actividades y entre los pasos.

Una línea horizontal en la que se juntan más de dos flechas representa que las actividades o los pasos anteriores se pueden realizar de manera independiente.

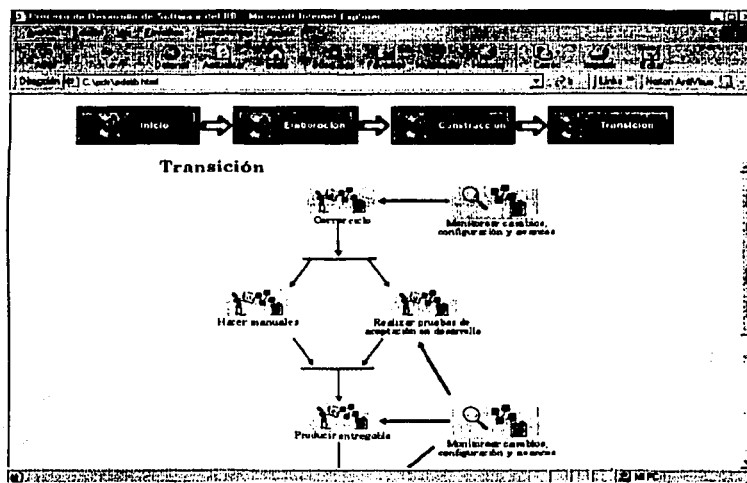
La primera pantalla de la herramienta muestra las cuatro fases del proceso: inicio, elaboración, construcción y transición. En esta pantalla inicial se muestra también los estándares para el ambiente de desarrollo del Departamento de Informática.

Enseguida se muestra la pantalla de inicio. Para entrar a cada opción se pulsa sobre los iconos.



Pantalla inicial de la herramienta.

En cada fase se despliega un diagrama que indica las actividades generales, para entrar a los detalles se pulsa en el icono deseado.



Ejemplo de un diagrama de actividades de una fase.

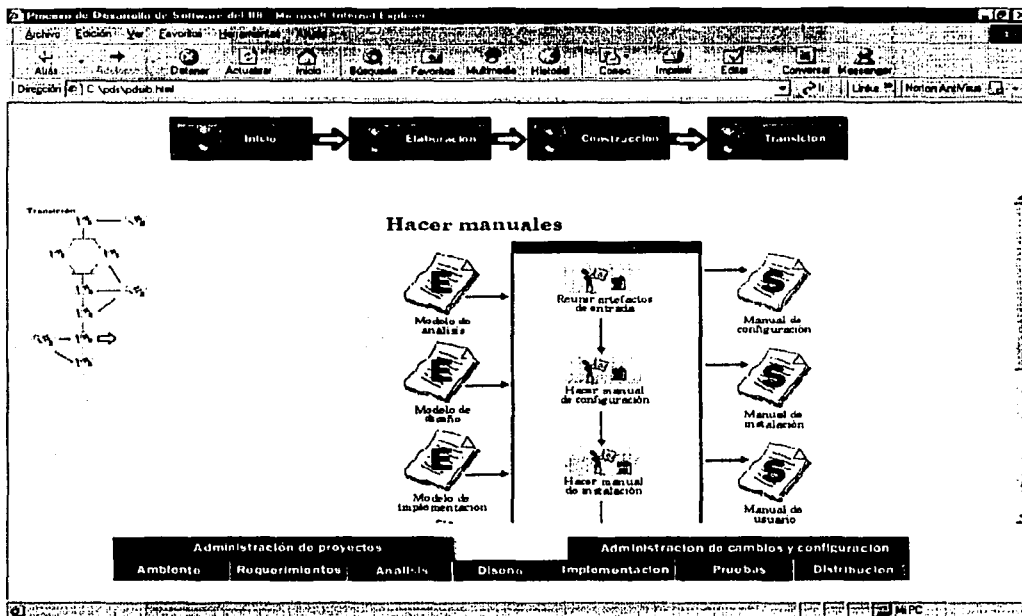
LEER CON
PALLA DE ORIGEN

TESIS CON FALLA DE ORIGEN

Cuando se pulsa sobre la actividad general, se despliega un diagrama de actividades detallado que indica los artefactos de entrada, el carril del rol responsable de la actividad y los artefactos de salida.

En la parte izquierda aparece el diagrama general de actividades de donde viene el diagrama detallado. Se puede navegar a través de ese diagrama también.

En la parte de abajo de la ventana aparecen los flujos de trabajo del proceso y se resalta el que tiene relación con el diagrama que se está mostrando.



Ejemplo de un diagrama de actividades detallado.

Con la construcción de esta herramienta se concluye este trabajo, el de adaptar los flujos de trabajo de ambiente, distribución y pruebas basados en el Proceso Unificado y el de crear el proceso de desarrollo de sistemas del Departamento de Informática del Instituto de Investigaciones Bibliográficas, obtenido al integrar estos tres flujos junto con los de [Sánchez] y [Ramírez].

Es importante mencionar que en los apéndices se encuentran las especificaciones a detalle de las actividades de los flujos de trabajo de ambiente, distribución y pruebas, tal cual, lo indica el Proceso Unificado de Rational.

Conclusiones

Los procesos de software ayudan a profesionalizar el área de ingeniería de software debido a que incorporan orden, sistematización, estructura, descripción, estandarización, reutilización, evolución, continuidad, confianza, especialización, guías y enseñanza para obtener productos de calidad.

Los procesos de software ayudan a asignar responsabilidades con mayor facilidad debido a que los roles marcan las actividades que se deben llevar a cabo, organizan el trabajo de tal forma que en cualquier momento se puede saber que se está haciendo y mantienen información acerca del proyecto para conocer como se está desarrollando.

Los procesos de software ayudan a disminuir la complejidad de las relaciones en las actividades involucradas durante el ciclo de vida de los proyectos, permiten controlar todas las actividades del desarrollo de productos y su mantenimiento.

Antes de adaptar cualquier flujo de trabajo del Proceso Unificado es necesario leer con detenimiento el flujo **ambiente** ya que éste indica las consideraciones para adaptar el proceso.

La adaptación se hizo con la idea de tomar las actividades importantes que el personal de desarrollo de sistemas esta dispuesto a hacer.

La adaptación del proceso llevada a cabo por las mismas personas que lo seguirán ofrece la ventaja de que no se toma como un proceso impuesto donde no se sabe porque se tomaron ciertas decisiones.

Debido a que los procesos de software evolucionan de forma natural es necesario incorporar en la descripción una forma fácil para hacer modificaciones.

El estudio de la organización a la que se le va a crear un proceso de desarrollo de sistemas es elemental para poder establecer las actividades, roles y artefactos de cada flujo de trabajo.

Para poder adaptar los flujos de trabajo del Proceso Unificado al Departamento de Informática fue necesario conocer incluso a las personas, la forma en que realizan las actividades de desarrollo, las herramientas de software que utilizan para crear los sistemas, el tipo de sistemas que desarrollan, los tipos de problemas relacionados al desarrollo e implementación de sistemas de software, el perfil de sus usuarios, etc.

Al hacer la adaptación del flujo de trabajo **ambiente** se crearon diferentes artefactos como el estándar del proceso de desarrollo de sistemas del IIB (PDSIIB), el estándar para modificar el proceso, el comité responsable del PDSIIB, la solicitud de cambio el PDSIIB, el estándar de herramientas, herramientas a comprar y la configuración de herramientas. De

TESIS CON
FALLA DE ORIGEN

TESIS CON
FALLA DE ORIGEN

estos artefactos el más difícil de hacer fue el "estándar para modificar el proceso" debido a que éste debe considerar con precisión la manera en que se deben hacer los cambios.

El flujo de trabajo **ambiente** quedó establecido con las actividades, conocer el ambiente para el proyecto y preparar el ambiente para un ciclo. Este flujo fue el más complicado de adaptar debido a que éste no incorpora las actividades para crear nuevos procesos de desarrollo de sistemas.

Al hacer la adaptación del flujo de trabajo **distribución** se crearon diferentes artefactos como el estándar de distribución, el estándar para los manuales del sistema, el plan de distribución, la invitación de pruebas formales, aceptación de pruebas en sitio, el informe de entregable y la aceptación de pruebas en sitio. Estos artefactos son de los más fáciles de hacer si se comparan con los artefactos de los otros dos flujos.

El flujo de trabajo **distribución** quedó establecido con las actividades hacer plan de distribución, hacer manuales, realizar pruebas de aceptación en desarrollo y producir entregable. Este flujo es el más fácil de adaptar al compararlo con los otros dos.

Al hacer la adaptación del flujo de trabajo **pruebas** se crearon diferentes artefactos como el estándar de pruebas, el plan de pruebas, el caso de prueba y el reporte de evaluación de pruebas. El artefacto más complicado de hacer fue el de "caso de prueba" ya que éste debe ser utilizado en todos los niveles y estrategias de prueba.

El flujo de trabajo **pruebas** quedó establecido con las actividades hacer el plan de pruebas, hacer el diseño de las pruebas, ejecutar las pruebas y evaluar las pruebas. Al hacer la adaptación de este flujo se presentó una dificultad intermedia al compararlo con el flujo de ambiente que fue el más difícil y el de distribución que fue el más fácil.

Los roles de cada flujo de trabajo no sufrieron ningún cambio y quedaron establecidos de la misma manera que el Proceso Unificado, lo cual indica que los roles fueron los más fáciles de adaptar al compararlos con las actividades y con los artefactos.

Para que la adaptación de las actividades y sus pasos tuviera cierta objetividad, se planteó que la descripción del proceso se hiciera con la idea de que lo utilizarán personas que no conocen al Departamento de Informática, lo cual resulta más difícil que si se hubiera considerado que al proceso únicamente lo van a utilizar las personas que ya conocen al departamento.

Al diseñar los artefactos se procuró que tuvieran una estructura de llenado rápido para promover la utilización de los mismos. Para ello, se puso en los artefactos cuando era posible, las opciones más comunes. Lo cual resulta complicado porque puede ser que esas opciones queden obsoletas en cierto momento y por ello se tengan que estar modificando los artefactos.

El planteamiento de crear una herramienta que se encuentre disponible en un sitio web se hizo con la intención de hacer más fácil el acceso a las descripciones de las actividades y de los artefactos.

Se creó la herramienta del PDSIIB con la finalidad de ofrecer un ambiente visual que facilite el seguimiento del proceso, debido a que está basada en diagramas. Hacer la herramienta es fácil cuando el proceso está completamente terminado, pero cuando no es así, resulta muy tedioso porque cada cambio hecho en algún elemento del proceso se debe actualizar en la herramienta.

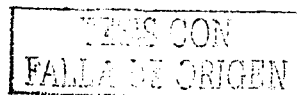
La limitación de recursos económicos y de poco personal no impide tener un proceso de software en una organización, sin embargo lo que sí puede ser una limitante es el desconocimiento de cómo crearlo. Al iniciar la adaptación de los tres flujos de trabajo se visualizaba una complejidad mayor que la que se presentó cuando se iba avanzando en el análisis de cada actividad y cada artefacto. Esta es una invitación a acercarse a los procesos de software.

Lo más complicado de este trabajo fue integrar estos tres flujos con los demás flujos adaptados de [Sánchez] y [Ramírez], el cual, se logró exitosamente. Para poder hacer la integración se tuvieron que escoger las actividades de cada flujo y colocarlas en alguna de las cuatro fases del proceso adaptado, las fases son inicio, elaboración, construcción y transición. Una vez colocadas las actividades dentro de alguna de las fases se realizó un seguimiento de las actividades, artefactos y roles, es decir, se hizo el seguimiento de la secuencia de actividades a través de las fases para comprobar que no había incongruencias. Para revisar que los artefactos se diseñaron correctamente se revisó en qué actividad se generaban y en qué otra actividad se volvían a utilizar para apoyar otras actividades o el llenado de otros artefactos. Y para los roles se hizo lo mismo.

Consideraciones futuras

Integrar herramientas de software de apoyo al proceso para enriquecer la facilidad que implica el uso de la automatización.

Crear el sistema de información de administración del proceso de desarrollo de software junto con un repositorio donde se guarden los productos de los proyectos y sus artefactos para proporcionar la manipulación completamente electrónica.



TEMS CON
FALLA DE ORIGEN

Apéndice A

Flujo de trabajo ambiente de acuerdo al Proceso Unificado de Rational

El propósito de este flujo es realizar las actividades para configurar el proceso, crear los artefactos acordes a la empresa de desarrollo de software y proporcionar soporte al equipo de desarrollo.

Actividades

Preparar el ambiente para el proyecto

Evaluar la organización actual de desarrollo

El propósito es describir el estado actual de la empresa de software en términos de su proceso actual, herramientas disponibles, competencias y actitudes de las personas, clientes, competidores, tendencias técnicas, problemas y áreas de mejora.

Para configurar un proceso es necesario entender el contexto de los proyectos; ése es el estado actual de la organización de desarrollo en términos de las personas, su proceso y las herramientas de apoyo.

Es importante entender las áreas de problema y las áreas potenciales para hacer mejoras, así como la información sobre los factores externos como los competidores y tendencias en el mercado.

Cuando éste paso está completo, se sabe:

- El estado actual de la empresa
- Los tipos de personas involucradas y sus niveles de competencia, habilidades y motivación
- Las herramientas actualmente usadas por la organización
- El software actual para el diseño del proceso y cómo se describe

Después de hacer la evaluación del estado actual se deberá:

- Utilizar el estado actual como entrada para configurar el proceso.
- Identificar las áreas que necesitan ser mejoradas primero. No se deberá introducir el proceso completo y todas las herramientas inmediatamente, se debe hacer en

TESIS CON
FALLA DE ORIGEN

incrementos y empezando con las áreas que tienen las mayores necesidades y el potencial para crear mejoras.

- Explicar a los patrocinadores por qué necesitan cambiar el proceso, herramientas y personas.
- Crear la motivación en las personas de la organización para considerar a quién se afecta directamente e indirectamente por los cambios.

Pasos

1. Comenzar la valoración

Se debe comenzar la valoración con un taller donde se perciban las personas importantes del proceso actual. Los propósitos del taller inicial son para que los ingenieros del proceso encuentren a los individuos que son afectados por el proyecto y hacer una lista de los problemas que tienen cada uno de ellos.

2. Identificar personas afectadas por el sistema

Identificar a las personas afectadas por la aplicación del proceso. Identificar a los afectados que se encuentran fuera de la empresa, como: Clientes, requerimientos de los clientes en términos de tiempo y mercado, rasgos de seguridad, robustez y complejidad;

Competidores, quiénes son los competidores, en qué áreas son fuertes los competidores, qué se puede aprender de los competidores;

Otros afectados, revisar si hay otras personas afectadas por el proyecto, como proveedores involucrados, revisar si las relaciones con ellos son un problema.

Identificar personas afectadas dentro de la empresa, como: miembros, administrador del proyecto, personas en otras secciones de desarrollo, personas de otros departamentos, preguntar que expectativas tiene la empresa.

3. Describir la organización interiormente

Describir como es la organización hacia el interior, particularmente los roles actuales y los equipos de desarrollo. También, mirar la relación entre las diferentes partes de la empresa. Por ejemplo, la relación entre el desarrollo y mantenimiento, y la relación entre el desarrollo y las pruebas.

4. Identificar a las personas importantes

Identificar a las personas importantes en la organización.

Una persona importante es alguien que tiene una o algunas de las características siguientes:

- Las masas lo escuchan
- Pueda actuar como un guía
- Es experto en alguna área
- Se opone a la aplicación del proceso
- Es responsable del presupuesto

Para tener éxito al llevar a cabo el proceso junto con sus herramientas es necesario tener a las personas importantes, después, cuando se lleve a cabo deberá involucrarseles durante el resto de la valoración para obtener información, como expertos para ayudar a ajustar el proceso, para trabajar en el proyecto piloto tomando el rol de guía. Deberá tener cuidado con las personas que quieren discutir el proceso en lugar del desarrollo de sistemas.

5. Identificar las razones para realizar cambios

Preguntar a los interesados en el sistema por qué quieren cambiar a un nuevo proceso y utilizar nuevas herramientas para llevar a cabo el Proceso Unificado.

Algunas respuestas típicas y sus efectos de como se lleva a cabo cierto proceso:

A) "Compramos herramientas y conseguimos el Proceso Unificado porque es de libre uso y parece ser que encierra buenas ideas." En este caso, se tiene que tener mucho cuidado, probablemente sólo se haga la instrumentación de pequeñas porciones del proceso.

B) "Hemos empezado a usar una nueva tecnología y guías que cubran las necesidades." En este caso, se puede querer introducir sólo las partes del proceso que guíen hacia usar la nueva tecnología.

C) "Necesitamos un proceso, pero no queremos desarrollarlo nosotros mismos, intentamos desarrollar nuestro propio proceso, pero realmente fue agobiado por el trabajo de desarrollo, no podríamos permitirnos el lujo de desarrollar y mantener el proceso." En este caso, es más agresivo llevar a cabo el proceso. Se han desarrollado y usado procesos antes, por consiguiente es probable tener una vista pragmática del proceso y su valor.

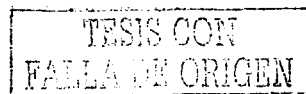
6. Identificar habilidades

Hacer una lista de áreas de competencia importantes como las siguientes:

- Proceso
 - Modelado del negocio
 - Requerimientos
 - Diseño de la interfaz gráfica de usuario
 - Análisis y diseño
 - Implementación
 - Lenguajes de programación, bases de datos relacionales, etc.
- Herramientas
- Dominio del problema
- Tecnología
 - Sistemas distribuidos, etc.,
- Habilidades administrativas y organizacionales
- Habilidades de comunicación e interpersonales
- Conocimiento organizacional

Para cada una de éstas áreas relevantes, evaluar el conocimiento, especialización y experiencia de las personas en la organización.

Hay varias maneras de hacer esto:



Pedir a uno o a varios gerentes que clasifiquen las competencias individuales o de equipos completos.

Pedir a los empleados que ellos mismos clasifiquen sus competencias por cada área.

Para cada área, estimar las competencias individuales. Enseguida se muestra un ejemplo de una balanza de competencia que puede ser usada:

- Experto --- Puede actuar como guía.
- Calificado --- Puede trabajar independientemente.
- Aprendiz --- Puede necesitar ayuda de vez en cuando.
- Novato --- Puede no estar familiarizado con el asunto.

Hacer un perfil de competencia por cada individuo y reúna los perfiles de competencia para los equipos completos. Esto es importante para entender el perfil de competencia de un equipo en conjunto, porque ningún individuo tendrá todas las competencias juntas.

7. Evaluar actitudes de las personas

Entrevistar a las personas para entender las actitudes hacia el cambio a nuevas tecnologías, nuevas herramientas y el nuevo proceso. Si las personas son negativas o escépticas al cambio, es imposible tener éxito al tratar de hacer un cambio, a menos que transforme las actitudes negativas en positivas.

8. Estimar la capacidad para hacer cambios

Analizar la capacidad de hacer cambios en la organización y entre los individuos. Al mirar los problemas de la organización, hay una tendencia a querer arreglar todo, especialmente porque muchos de estos problemas ocurren juntos. Las organizaciones así como los individuos, sólo puede acomodarse dentro de un rango limitado. Existen organizaciones que están mejor preparadas para el cambio que otras. Para entender la capacidad de la organización para enfrentar el cambio, entrevistar a las personas para entender las actitudes y su disposición a cambiar.

9. Analizar la descripción del proceso de desarrollo

Leer las descripciones del proceso existentes para entender la profundidad del proceso de desarrollo existente. Para cada proceso flujo de trabajo, identifique qué tipos de descripción tienen. Por ejemplo, averiguar si no hay descripciones de todo y si hay ejemplos, guías, plantillas de documentos, descripciones de actividades, descripciones de artefactos y así sucesivamente.

Con una buena comprensión del proceso existente se deberá:

- Entender qué tipo de descripción de proceso usan, si ya tienen una descripción del proceso, o probablemente desean adoptar el Proceso Unificado, sin embargo, si no han

tenido alguna descripción de procesos anteriormente, debe tener mucho cuidado para llevar a cabo el Proceso Unificado en pequeños incrementos.

- Entender qué áreas se necesita más de una descripción en el nuevo proceso.
 - Entender qué partes de la descripción del proceso puede seguir utilizando. Aun cuando la última meta es usar el Proceso Unificado completo, se llevará a cabo en incrementos, esto significa que el proyecto continúa usando alguna de las descripciones del proceso existente durante algún tiempo.
10. Caracterizar el proyecto y la aplicación

Describir las características de proyectos y aplicaciones típicas de la organización. Visualizar proyectos terminados y proyectos que actualmente están en producción. Reúna información entrevistando a cada uno de los individuos que pertenecen a los diferentes equipos. Estudiar los artefactos resultantes de proyectos anteriores tales como planes, descripciones de procesos existentes, valoraciones de estado finales anteriores, etc.

Reunir las características siguientes para cada proyecto y producto

- Tamaño del esfuerzo de desarrollo de software. ¿Cuál es el tamaño del producto?
- Grado de novedad. ¿En que fase, en una balanza entre fase de desarrollo y mantenimiento está el producto?
- Tipo de aplicación. ¿Es crítica la aplicación? ¿Hay algún requerimiento para seguir normas específicas?
- Identificar qué tipo de desarrollo se hace en la organización. Por ejemplo, si se contratan servicios de desarrollo, si se hacen desarrollos internos, etc.
- Complejidad técnica. ¿Cuáles son los problemas técnicos y desafíos al construir los productos? Por ejemplo, ¿Es la aplicación un sistema distribuido, un sistema de seguridad crítico o un sistema de alta integridad? ¿Hay un sistema que será reusado?

11. Identificar las herramientas de apoyo

Identificar las herramientas que se usan actualmente en los proyectos e igualmente las áreas donde hacen falta herramientas de apoyo.

12. Identificar problemas

La mejor manera de identificar problemas es reunir a varias personas importantes en una sesión. Note que hay situaciones en que no tiene sentido tratar de identificar problemas puesto que no hay una forma de trabajo definida.

Para identificar problemas, se deben hacer preguntas como:

- ¿Cuáles son los problemas de los proyectos?
- ¿Hay alguna percepción de que algo no anda bien?
- ¿Los proyectos ordinariamente se entregan fuera de tiempo o por encima del presupuesto?
- ¿Tienen métricas que puedan ser analizadas?

TESIS CON
FALLA DE ORIGEN

Asegurarse de cubrir todas las áreas de los procesos de desarrollo, incluso todos los flujos de trabajo, herramientas de apoyo y competencias. También se deben visualizar los problemas organizacionales y políticos.

Identificar que efectos negativos tiene o tendrá cada problema, si no se elimina o se reduce, conozca como el efecto de un problema le ayuda a entender que tan crítico es eliminarlo o reducirlo.

Identificar la raíz de las causas de cada problema para ayudarlo a entender cómo quitarlas o reducirlas y que gastos implican.

Clasificar los problemas con respecto al efecto que ellos causan. Por ejemplo, usar una escala del uno al cinco, donde cinco se utiliza para los problemas más críticos y uno es para los problemas menores. El propósito principal es entender la importancia relativa de los problemas.

Tabla de problemas (Ejemplo)

Problema	Efecto	Causas	Clasificación
La calidad del software que se libera es mala.	- Los clientes quedan insatisfechos. - Tenemos que arreglar defectos después de hacer la entrega principal.	- Los casos de prueba no proporcionan una cobertura amplia. - Las pruebas no están automatizadas. - Los ingenieros de pruebas no tienen la capacitación adecuada.	5
...

13. Escribir las conclusiones

Analizar los resultados de la información colectada y reúna una lista de áreas y problemas en los cuales enfocarse. Problemas que deben ser regidos por algunas de las categorías siguientes:

- Áreas de problema mayores, donde puede mejorar significativamente el desempeño de los proyectos.
- Áreas donde puede conseguir beneficios a corto plazo y donde puede mostrar resultados rápidamente.
- Áreas donde una mejora tendrá visibilidad alta.

Documentar la información obtenida y las conclusiones en el artefacto "Valoración de la organización de desarrollo".

14. Hacer recomendaciones

Frecuentemente es necesario hacer algunas recomendaciones para el futuro como parte de la valoración. La recomendación debe describir cómo el proyecto de desarrollo o la empresa debe llevar a cabo el nuevo proceso y uso de nuevas herramientas. Las

recomendaciones podrían consistir en planes de aplicación esbozados, planes de proyectos, casos de desarrollo y otros más.

Delinear el Caso de Desarrollo.

El propósito es describir el proceso de desarrollo de software para un proyecto o proyectos de la organización.

Pasos

1. Analizar el proyecto y la organización

De la valoración de la organización de desarrollo se obtiene la información sobre el proyecto y la organización. Analizar los diferentes aspectos para conocer cómo afectan al caso de desarrollo.

Definir el alcance

Definir qué flujos de trabajo debe cubrir el caso de desarrollo, cuidando los aspectos considerados en la valoración de la organización de desarrollo.

Se recomienda que se lleve a cabo el proceso en varias iteraciones. El caso de desarrollo no tiene que capturar el proceso completo.

3. Decidir cómo realizar cada flujo de trabajo

Una vez que se ha decidido que flujos de trabajo se van a introducir se debe decidir lo siguiente:

- Cómo realizar el flujo de trabajo.
- Qué partes del flujo de trabajo se utilizarán.
- Cuándo, durante el ciclo de vida del proyecto se introducirán los flujos de trabajo y sus partes.

Cuando se desea introducir un flujo de trabajo o parte de éste hay que tomar en cuenta lo siguiente:

- Pertinencia. ¿Es aplicable para el proyecto? ¿Realmente agrega valor al introducirlo?
- Problemas y causas. ¿Se relaciona con problemas percibidos y sus causas?
- Herramientas de apoyo. ¿Qué herramientas de apoyo son necesarias?
- Tiempo. ¿Cuándo durante el ciclo de vida del proyecto debe introducirse?
- Costo de implementación. ¿Cuál es el costo de implementación en el proyecto? Esto incluye:
 - Costo para capacitar a los miembros del proyecto.
 - Costo para instalar las herramientas de apoyo.
 - Costo para desarrollar guías y plantillas.

TESIS CON
FALLA DE ORIGEN

4. Ajustar los artefactos por cada flujo de trabajo

Ajustar los artefactos para cada uno de los flujos de trabajo.

Realizar los siguientes pasos:

Decidir como deberá ser utilizado el artefacto

Clasificación	Explicación
Debe hacerse	Se debe usar el artefacto. Es un artefacto importante y puede causar problemas posteriores en el desarrollo si no es producido
Debería hacerse	Se debería tener este artefacto, si es posible, pero es negociable. Si no se produce debe haber una justificación.
Podría hacerse	Significa que el artefacto no será producido. Solo se produce si agrega valor y si sobra bastante tiempo.
No se hará	Significa que no se producirá el artefacto. Esto ocurre cuando ya existe un artefacto parecido y reemplaza al del Proceso Unificado.

- Decidir el nivel de revisión por cada artefacto
- Decidir cómo debe capturar los resultados finales de un flujo de trabajo. ¿Necesita guardar los resultados en papel? En ese caso, tiene que decidir si es uno o varios informes que extraen los resultados de las herramientas e imprime los resultados en papel.
- Decidir qué herramientas usará para desarrollar y mantener el artefacto.
- Decidir qué propiedades usará y cómo usarlas.
- Cuando es importante, decida qué estereotipos usará.
- Hacer un bosquejo para generar artefactos.

Determinar si es necesario algún reporte de trabajo para extraer información de los modelos y documentar la información sobre papel para hacer las revisiones. Estos reportes de trabajo son usualmente temporales y se desecharán cuando la revisión esté completa.

5. Modificar los flujos de trabajo y actividades

Estudiar el conjunto modificado de artefactos y las actividades, preparar y actualizar los artefactos. Determinar cuando se deberá modificar o simplificar estas actividades. Asegurarse de descartar algún paso o actividad innecesaria.

Considerar lo siguiente:

- ✓ Introducir nuevos pasos y posiblemente nuevas actividades para reflejarlas en los artefactos, reportes y documentos que se han tenido que agregar.
- ✓ Examinar cómo las herramientas utilizadas pueden facilitar, automatizar o incluso puede suprimir algunos pasos.
- ✓ Introducir guías específicas y reglas para la organización, pueden agregarse puntos clave, puntos de control, elementos para revisión o dejarlos como documentos separados que pueden ser consultados posteriormente.
- ✓ Una vez que las actividades son conocidas, vuelva a revisar los flujos de trabajo que muestran cómo interaccionan las actividades, quitando o agregando actividades si es necesario.
- ✓ Pueden omitirse o crearse todo un flujo de trabajo.
- ✓ Puede tener que introducir algunos roles adicionales para actividades especiales.

Describir los cambios en el caso de desarrollo.

6. Escoger el modelo del ciclo de vida

Escoger el tipo de modelo de ciclo de vida que se empleará para el proyecto. Refinar el modelo del Proceso Unificado y ajustar los hitos y los criterios de evaluación de los hitos si es necesario.

7. Identificar a los afectados

Se necesita identificar y describir los diferentes tipos de afectados, sus necesidades y responsabilidades. Ejemplos de diferentes afectados son clientes representativos, usuarios representativos, inversionistas, gerente de producción y comprador, entre otros.

Se debe describir a los afectados y sus necesidades en el Caso de Desarrollo.

8. Perfilar los roles a los puestos de trabajo

En algunas organizaciones de desarrollo hay puestos de trabajo ya definidos. Si éstos puestos de trabajo son usados comúnmente y tienen una amplia aceptación en la organización, puede merecer la pena perfilar los roles del proceso hacia los puestos de trabajo de la organización. Perfilar los roles ayuda a las personas a entender que los roles no son puestos de trabajo.

Documentar estas relaciones en el Caso de Desarrollo.

9. Documentar el Caso de Desarrollo

Describir el Caso de Desarrollo. Es recomendable que se describa el Caso de Desarrollo en una o varias páginas web con hipervínculos en línea.

TESIS CON
FALLA DE ORIGEN

10. Mantener el Caso de Desarrollo

Muchas de las decisiones deben hacerse antes de que el proyecto termine. Después de cada iteración en el proyecto, se debe evaluar el proceso y revisar las decisiones que se han tomado. Si una nueva versión del Proceso es liberada se necesitará poner al día el caso de desarrollo.

Producir una lista de herramientas candidato a usar en el desarrollo.

Pasos

1. Identificar necesidades y restricciones

Identificar cuáles son las necesidades de herramientas de apoyo y cuáles son las restricciones, tomando en cuenta lo siguiente:

- El proceso de desarrollo. Qué herramientas de apoyo se requiere para trabajar eficazmente. Por ejemplo, si la organización decide emplear un proceso de desarrollo iterativo es necesario automatizar las pruebas, porque se harán pruebas muchas veces durante el proyecto.
- Plataformas de desarrollo.
- Plataformas designadas.
- Lenguajes de programación a utilizar.
- Herramientas existentes. Evaluar algunas herramientas existentes para decidir si se continuarán usando.
- La distribución de la empresa. ¿Está bien distribuida la organización? Las herramientas de desarrollo generalmente apoyan a diferentes distribuciones de una organización.
- El tamaño del esfuerzo de desarrollo. Las herramientas apoyan muy bien a organizaciones grandes.
- Presupuesto y restricción de tiempo.
- La valoración de la organización de desarrollo proporcionará una buena entrada.

2. Colectar información sobre herramientas

Reunir información sobre las herramientas candidato que se consideren adecuadas para el problema, recordando que no se puede quitar un tornillo con unas pinzas ni una tuerca con un desarmador e información sobre sus vendedores. Parte de esta información son datos que pueden colectarse preguntando directamente al vendedor o de alguna otra fuente.

Características y funciones de las herramientas

Realizar una lista de características y funciones de la herramienta que se está estudiando. En la mayoría de los casos los vendedores proporcionan tales listas. La tabla siguiente muestra un fragmento de una lista para herramientas de administración de configuración.

Características y funciones
Versiones de todos los objetos del sistema de archivos
Directorios para las versiones
Mezcla de tipos de archivo
Compresión de archivos de texto y archivos binarios
...

Criterios para las herramientas

Criterio de la herramienta	Comentarios
Características y funciones	La funcionalidad que provee la herramienta. Esta deberá ser la conclusión global de esta tabla.
Integración	El nivel de integración con otras herramientas. ¿Cómo se transfiere la información entre las diferentes herramientas?. ¿Qué tan bien se ajusta a las herramientas existentes y a las otras herramientas que se están evaluando?. El nivel de integración frecuentemente es una característica muy importante. Las herramientas bien integradas son fáciles de usar.
Pertinencia	¿Qué tan bien apoya la herramienta al proceso de desarrollo?. ¿Tiene que cambiar la manera en que trabaja para usar la herramienta? La falta de pertinencia significa que tiene que cambiar la manera en que trabaja actualmente. Esto puede ser aún atractivo si es que la herramienta tiene otras fortalezas.
Extensibilidad	La habilidad de extenderse y personalizar la herramienta. La extensibilidad es buena porque significa que puede adaptar la herramienta a sus necesidades. Sin embargo, se debe de asegurar que no toma demasiado tiempo configurar la herramienta.
Apoyo del equipo	La habilidad de apoyar un equipo de usuarios. Conocer si la herramienta de apoyo se puede usar para equipos que están distribuidos geográficamente.
Usabilidad	La facilidad de aprendizaje y uso de la herramienta. Conocer el tiempo que toma el aprender a ser productivo con el uso de la herramienta. ¿Es la herramienta conveniente

TRABAJOS CON
FALLA DE ORIGEN

**TESIS CON
FALLA DE ORIGEN**

	para las personas que lo usarán frecuentemente? Asegúrese de analizar las funciones de uso más común. El hecho de que algunas funciones complejas se usen de vez en cuando provoca que sean ignoradas.
Calidad	Dependiendo del tipo de herramienta, la calidad de la herramienta determinará la calidad del producto que se está construyendo. La calidad es importante, especialmente si tiene impacto directo en el producto que se desarrollará. Por ejemplo, un compilador que produce código lento o un editor de comandos que produce errores frecuentes.
Desempeño	La efectividad total de la herramienta incluyendo la capacidad, accesibilidad y tiempos de respuesta. El mal desempeño puede ser aceptable si afecta funciones o capacidades que raramente se usan.
Madurez	El nivel madurez de la herramienta. Algunas organizaciones no comprarían una herramienta versión 1.0 de un nuevo vendedor sin tener en cuenta que tan buena es la herramienta.
Criterios del vendedor	Comentarios
Estabilidad	Arriesga su futuro basado en el futuro del vendedor. ¿Cuánto tiempo ha estado la empresa en el negocio? ¿Qué tan estable es la empresa? ¿Están invirtiendo la empresa en la herramienta? ¿La herramienta es la línea principal de la empresa, o es una herramienta secundaria?
Disponibilidad de apoyo	¿Qué apoyo está disponible a dar el vendedor y los socios potenciales? Puede necesitar ayuda para instalar y configurar la herramienta y apoyo continuo para los usuarios finales.
Disponibilidad para capacitación	¿Qué disponibilidad para capacitar tiene el vendedor y los socios potenciales?
Perspectivas de crecimiento	¿Qué tan bien apoya la herramienta el desarrollo de software?. Considere qué perspectivas tiene el desarrollo de software. ¿Que perspectivas tiene la herramienta de apoyo y que otras direcciones se podrían tomar?

Costo

Los costos asociados con adquirir y poseer la herramienta, incluye el costo de adquisición, el costo de implementación y el costo de mantenimiento. Decidir cuántos usuarios tiene y para cuánto tiempo quiere calcular el costo.

Costo	Comentario
Costo de adquisición	El costo de comprar la herramienta
Costo de implementación	El costo para tener la herramienta instalada e integrarla con el ambiente de desarrollo existente. Esto incluye el costo de capacitación a los usuarios de la herramienta, los usuarios finales y las personas que administrarán la herramienta.
Costo de mantenimiento	El costo continuo por el uso de la herramienta. Esto incluye el costo de administrar la herramienta, las actualizaciones y los costos cotinuos de capacitación para las personas que administran la herramienta y los usuarios finales de la herramienta.

3. Comparar diferentes herramientas

Combinar los factores y seleccionar las herramientas más buenas no es un problema trivial. Para ayudarle a tomar una decisión es recomendable que haga una lista de características.

Comparar características y funciones

Usando la lista de características y funciones decidir que tan importante es cada una de ellas. Usar la clasificación jerárquica siguiente:

- Debe --- La herramienta debe tener esta característica.
- Bueno --- Sería bueno que tuviera esa característica pero no es crítico.
- No requerido --- No importa si la herramienta tiene o no esa característica.

Indicar para cada herramienta si tiene la característica o no usando los símbolos siguientes:

Símbolo	Descripción
+	Tiene la característica
-	Falta la característica

4. Seleccionar herramientas

Seleccionar las herramientas que cubran mejor sus necesidades y pueda ajustarse a las restricciones. No caer en la trampa de comparar solo características y funciones. El otro criterio es igualmente o más importante. A menos que la opción de la herramienta sea obvia

se recomienda que pruebe la herramienta que ha encontrado acorde a sus necesidades antes de decidir adquirirla.

Si hay alguna duda sobre la herramienta, lo mejor es probarla antes de adquirirla o tratar de aplicarla al caso real. También puede intentar encontrar otras empresas que estén usando la herramienta para preguntales si ellos pueden evaluar la herramienta. También puede pedirles una referencia a los clientes de los vendedores, otros clientes que están usando la herramienta. También hay información disponible en Internet, por ejemplo, las revistas en línea.

Es importante saber que cambiar una herramienta en medio de un proyecto es en toda ocasión altamente costoso tanto en lo económico como en lo profesional, por ejemplo el profesional puede perder su trabajo.

5. Adquirir herramientas

Adquirir herramientas es un problema no trivial que involucra asuntos legales así como asuntos financieros.

Consideraciones:

- Uso. ¿Para que quiero una herramienta?
- Instalación. ¿Cuánta ayuda ofrecen para instalar las herramientas?
- Apoyo. ¿Qué tipo de apoyo ofrece el vendedor? Muchos vendedores ofrecen varios grados de calidad a escoger.
- Compromiso del vendedor. ¿Cómo se compromete el vendedor? ¿Si se encuentra con problemas qué tipo de ayuda pueden ofrecerle? ¿En qué tiempo y a qué costo?
- Influencia. ¿Qué influencia tendrá en el futuro la herramienta? ¿Cómo priorizará sus necesidades?
- Mantenimiento. ¿Cómo maneja el vendedor los errores en la herramienta?
- Capacitación. ¿Qué tipo de capacitación ofrece? ¿Cuál es la disponibilidad de cursos?
- Futuro del producto. ¿Hay un plan que describe el desenvolvimiento futuro de la herramienta?
- Licencias. ¿Debe comprar una licencia del sitio para todos los miembros del proyecto, o debe comprar una herramienta por cada individuo? Algunas herramientas ofrecen licencias que tienen un límite de usuarios en la organización.

Producir las plantillas específicas para los artefactos importantes del proyecto

El caso de desarrollo define qué artefactos y reportes deben usarse junto con la información de cómo personalizarlos a las necesidades del proyecto. Generar plantillas para todos los artefactos y reportes de acuerdo a las necesidades del proyecto.

Preparar el ambiente para una iteración

Delinear el Caso de Desarrollo

El propósito es describir el proceso de desarrollo de software para un proyecto o proyectos de la organización. Esta actividad es igual a la que se describió en preparar el ambiente para el proyecto.

Configurar herramientas

Pasos

1. Instalar la herramienta

Identificar qué otro software se requiere para la herramienta. Por ejemplo, una herramienta puede requerir un sistema de administración de base de datos primero.

Cuando se ha instalado el software de apoyo, se debe instalar la herramienta en el servidor.

2. Personalizar la herramienta (en el servidor)

Determinar cómo personalizar la herramienta para que apoye el proceso de desarrollo de la mejor manera. El caso de desarrollo servirá como una entrada.

Además de personalizar las herramientas se debe preparar los grupos de usuarios y derechos de acceso al servidor, en algunos casos la herramienta puede proveer sus propios mecanismos para ello. En otros casos, se definen grupos del usuario y derechos de acceso usando el sistema operativo. La configuración de grupos del usuario y derechos de acceso afectan cómo las herramientas pueden usarse. Por ejemplo, se podrán poner restricciones en las partes del almacén para que ciertos usuarios tengan acceso.

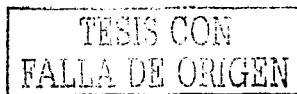
3. Configurar el soporte multisitio

Si el equipo está distribuido geográficamente puede ser necesario dividir el almacén en varios sitios o duplicar el almacén. Dividir o duplicar un almacén requiere que los almacenes en los diferentes sitios estén sincronizados, por lo tanto, la información debe transferirse entre los diferentes sitios.

4. Integrar con otras herramientas

Integrar la herramienta con otras herramientas. Una integración entre herramientas en muchos casos es la forma de extender una o varias herramientas. Una extensión de integración de una herramienta debe:

- Sincronizar datos entre las diferentes herramientas. Automatizar la creación y mantenimiento de elementos relacionados en proyectos de desarrollo.
- Permitir al usuario rastrear los elementos de diferentes herramientas.



- Permitir al usuario navegar entre las herramientas. Por ejemplo, acceder a un elemento en una herramienta de prueba de una herramienta de administración de requerimientos.
- Permitir al usuario ejecutar cierta funcionalidad de una herramienta. Por ejemplo, la posibilidad de crear elementos en otra herramienta.
- Permitir al usuario de herramientas de administración de versiones, manejar elementos desde una herramienta de administración de configuración.

5. Instalar y personalizar las herramientas en las computadoras cliente

Personalizar la herramienta en los clientes, así como se personalizó la herramienta en el servidor:

En algunos casos no tiene que hacer nada con el cliente. Por ejemplo, si el cliente tiene una interfaz web es suficiente con tener disponible la dirección de la aplicación en el servidor. Algunas herramientas le permiten hacer todo al hacer la personalización en del lado del servidor. Cuando los usuarios accesan al almacén en el servidor automáticamente obtienen la configuración correcta. Puede ser necesario instalar el software de integración en el cliente. Ponga el software de integración en un servidor y permita que los usuarios lo descarguen y lo instalen en sus computadoras.

Si es posible se debe automatizar la instalación de la herramienta y personalizar la herramienta para los usuarios. El beneficio de crear programas de instalación es que le permite preparar las herramientas para que los clientes consigan configuraciones correctas, extensiones y conexiones al almacén. Se debe crear la instalación de los programas y ponerlos en el servidor. Entonces los usuarios descargan los programas y los ejecutan para instalar la herramienta en su computadora.

Verificar la instalación y configuración de herramientas

Pasos

1. Verificar el ambiente
2. Verificar las herramientas
3. Verificar los datos
4. Ejecutar las herramientas

Producir las plantillas específicas para los artefactos importantes del proyecto.

El caso de desarrollo define qué artefactos y reportes deben usarse junto con la información de cómo personalizarlos a las necesidades del proyecto. Generar plantillas para todos los artefactos y reportes de acuerdo a las necesidades del proyecto.

Lanzar el caso de desarrollo

Pasos

1. Hacer los cambios públicos

Informar a todos los miembros del proyecto sobre el caso de desarrollo, guías, plantillas y herramientas. Si el proyecto tiene un sitio web, se notifica a los miembros del proyecto.

2. Capacitar a los miembros del proyecto

A menos que el cambio sea trivial, se necesita educar a los miembros del proyecto sobre el nuevo caso de desarrollo, guías, plantillas y herramientas.

Preparar las guías del proyecto para una iteración

El propósito es preparar o actualizar las guías necesarias para una iteración respecto al estilo del modelado del negocio, la interfaz de usuario, el lenguaje de programación, el diseño, los manuales, las pruebas y herramientas.

Apoyar el ambiente durante la iteración

El propósito es apoyar a los desarrolladores en el uso del proceso y herramientas en una iteración. Se debe apoyar con servicios como mantenimiento de la infraestructura de desarrollo para hardware y software, administración del sistema, respaldos, telecomunicaciones, generación de documentos y su reproducción, etc.

TESIS CON
FALLA DE ORIGEN

Apéndice B

Flujo de trabajo distribución de acuerdo al Proceso Unificado de Rational

El propósito de este flujo es asegurar que el sistema estará disponible para los usuarios finales.

Actividades

Desarrollar el plan de distribución

Propósito

Hacer el plan de distribución del producto que especifica las necesidades que deberán tomarse en cuenta para definir cómo y cuándo el producto estará disponible para el usuario final, para asegurar el éxito de la transición al nuevo sistema además de la manera de llevar a cabo las actividades del negocio. El plan de distribución se enfoca a la entrega del software junto con la elaboración del material de apoyo y de capacitación acerca del sistema para asegurar que los usuarios finales puedan usarlo.

Pasos

1. Plan para producir el software

Los programas ejecutables deben asociarse con otros artefactos para constituir la unidad de distribución completa.

- Documentación de instalación
- Documentación de usuario
- Datos de configuración
- Programas adicionales para migración: conversión de datos.

En algunas circunstancias, los programas ejecutables son producidos para diferentes configuraciones de usuario o diferentes conjuntos de artefactos. Para el software distribuido, los diferentes conjuntos son producidos para los nodos de la red.

TESIS CON
FALLA DE ORIGEN

2. Plan para empaquetar el software

Se empaquetan los artefactos que constituyen el producto a entregar en medios de comunicación convenientes: disquetes, cintas, discos compactos, archivos de servidor, libros, videos, etc., debidamente etiquetados. Frecuentemente se involucran otras organizaciones externas para empaquetar el software.

3. Plan para distribuir el software

Hay una amplia gama de nuevas opciones para distribuir el software, servicios de envío de paquetería, usar la red de los distribuidores o la distribución a través de Internet.

Es importante controlar quién está autorizado para usar el software, es decir, el uso de licencias, que normalmente involucran procedimientos y herramientas para administrarlas.

4. Plan para instalar el software

Cada vez más con el advenimiento de distribución a través de Internet, la instalación del software es un proceso de usuarios controlado. Debe ser apoyado por herramientas de instalación y procedimientos entregados junto con el producto. En algunos casos raros, la instalación es realizada por el vendedor del software.

En el caso de un sistema distribuido la instalación es generalmente más compleja, porque todos los nodos tienen que estar conectados en el momento oportuno y además la instalación se hace en procedimientos de instalación múltiple.

5. Migración

Como parte de la instalación el problema de migración viene frecuentemente:

- Reemplazando un sistema más viejo con uno nuevo, con o sin restricciones de continuidad de funcionamiento.
- Convirtiendo datos existentes a un nuevo formato.

Se desarrollan los programas asociados con la migración.

6. Proporcionar ayuda y asistencia a los usuarios

Esto puede tomar varias formas:

- Cursos de capacitación formales.
- Capacitación basado en computadoras
- Guías y ayuda en línea
- Soporte telefónico
- Soporte vía Internet
- Complementarios: pistas, notas, ejemplos, etc.

El apoyo involucra establecer procedimientos para rastrear problemas junto con la solución, esto es contemplado a través de la administración de cambios.

Preparar la factura

Propósito

Crear una lista completa de artefactos que van a constituir el producto. La lista incluye artículos de configuración de software, documentos y archivos de instalación. En el caso de productos empaquetados la factura de materiales necesarios que constituyen el producto final.

Pasos

1. Listar todos los artículos a entregar

El administrador de distribución necesita asegurarse que todos los elementos que componen el producto se encuentran listados en la factura. El plan de desarrollo de software recoge toda la información para manejar el proyecto.

2. Actualizar la factura de materiales

Todos los artículos serán listados en la factura de materiales y deben ponerse al día en cada iteración.

Desarrollar el material de apoyo

Desarrollar el material para la capacitación

Propósito

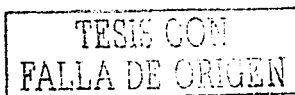
Producir el material necesario para capacitar a los usuarios.

Pasos

1. Desarrollar un bosquejo de los materiales para la capacitación

Determinar el público a capacitar. Qué tipo de roles tienen las personas típicas de la organización. Qué requisitos deben cubrir aquéllos que asistan al curso.

Definir objetivos para la capacitación. ¿Sobre qué rasgos específicos del producto deben aprender los estudiantes? En muchos casos no es posible encajar todo los detalles del uso de un producto en una sesión de capacitación. ¿Qué nivel de conocimiento deben lograr los estudiantes? Algunas cosas pueden ser suficientes discutir a un nivel de introducción, mientras otros son críticos y necesita asegurar que los estudiantes se sientan cómodos para aplicar los conocimientos después del curso.



Discutir qué tipo de capacitación sería conveniente. Las opciones a considerar son:

- Tutoriales en línea
- Capacitación en salones de clase
- Capacitación a través de talleres
- Decidir la duración de los cursos y el número de participantes.

Determinar cómo deben hacerse los materiales

2. Escribir los materiales para la capacitación

Los principales componentes de los materiales para capacitación se enfocan a la clase de estudiantes que los usarán en clases. Si se están escribiendo manuales para salones de clase o para talleres deberán incluir diapositivas y material para hacer ejercicios. Se deberán crear pautas para los instructores si es necesario.

Dependiendo del tema de clase, también puede ser útil incluir una lista de referencias a libros, artículos, sitios de internet, etc., que tienen información relacionada. En algunos casos puede escoger incluir reimpresión de los materiales de referencia.

Para asegurar una buena calidad de los materiales puede involucrar a un asesor para repasar los materiales y discutir problemas particulares. Se podría incluir a los representantes de los usuarios finales y a desarrolladores del producto.

Desarrollar los materiales de apoyo

Escribir materiales para el usuario final involucra las siguientes prácticas:

- Organizar la información para la facilidad de acceso.
- Escribir instrucciones fáciles de seguir.
- Estructurar la información fácil de examinar, mantenga información suficiente para los novatos.
- Diferenciar los tipos de información como conceptos, propósitos, experiencias y acciones.
- Apoyar el texto con gráficos para aclarar conceptos e instrucciones.
- Las explicaciones deben ser apoyadas por ejemplos para ayudar a los usuarios a aplicar la información a su contexto.

Los materiales de apoyo para los usuarios finales también pueden estar contenidos en sitios web. Producir este tipo de materiales es en sí mismo un proyecto que puede manejarse en iteraciones.

- Producir un plan de alto nivel para la información que será presentada.
- Especificar los contenidos a detalle.
- Implementar el contenido de la información.
- Probar y producir los materiales.
- Evaluar los materiales.

Administrar las pruebas de aceptación

Propósito

Para asegurar que el producto cumple los criterios de aceptación en el sitio de desarrollo y en el sitio de instalación designado.

Pasos

1. Asegurar la disponibilidad de las pruebas formales

Las pruebas de aceptación son la comprobación formal para determinar si el sistema satisface los criterios de aceptación. Se realizan en el sitio de desarrollo para después hacerlas en el sitio de producción.

El cliente evalúa los artefactos para determinar si cumplen con los criterios de aceptación, esto se describe en el plan de aceptación del producto el cual describe los roles, responsabilidades, recursos, tareas, casos de prueba y criterios de aceptación.

Otra entrada útil para determinar si el proyecto cumple con las pruebas formales es la auditoría de configuración en donde se informa:

- La concordancia del software desarrollado conforme a sus requerimientos.
- Los artefactos requeridos.

El administrador de distribución debe repasar las deficiencias menores con el cliente para saber si las pruebas continuarán.

El administrador del sistema necesita asegurar que la infraestructura y el software de apoyo está listo para las próximas actividades de pruebas.

2. Inicio de las pruebas

Una vez que el administrador de distribución está satisfecho con respecto a la disposición de todos los elementos que serán probados se continúa con el calendario de las pruebas de acuerdo al plan de aceptación del producto. Las pruebas comienzan cuando todos los participantes están presentes.

El probador carga todo el software y los archivos de datos de prueba y procede con la comprobación de los casos de prueba, procedimientos de prueba y los resultados esperados en el modelo de pruebas. Se guardan los resultados junto con cualquier diferencia entre los resultados esperados y los reales.

3. Registrar los resultados de la revisión y anomalías

Deben repasar los resultados de las pruebas al final del día. En esta reunión, los resultados para cada caso de prueba necesitan ser revisados y anotarse si fueron aceptados o no. Las minutas de la reunión necesitan guardarse sobre cualquier asunto importante y cualquier tipo de decisión.

En caso de que los resultados de las pruebas sean considerados como inaceptables el administrador de distribución puede hacer solicitudes de cambio.

TESIS CON
FALLA DE ORIGEN

Ejecutar las pruebas

Propósito

Ejecutar las pruebas y anotar los resultados de las pruebas.

Pasos

1. Ejecutar los procedimientos de prueba

Para ejecutar las pruebas, deben seguirse los siguientes pasos:

- Estructuración del ambiente de pruebas para asegurar que todos los componentes necesarios como hardware, software, herramientas, datos, etc. han sido implementados y están en el ambiente de pruebas.
- Inicializar el ambiente de pruebas para asegurar que todos los componentes está en el estado inicial correcto para comprobación de resultados.
- Ejecutar los procedimientos de prueba.

Nota: La ejecución de los procedimientos de prueba varían dependiendo si son automatizados o manuales.

2. Evaluar la ejecución de las pruebas

La ejecución de pruebas finales termina con alguna de dos condiciones:

- Normal: todos los procedimientos de prueba ejecutados. Si las pruebas terminan normalmente entonces se continúa con la verificación de resultados.
- Anormal: Los procedimientos de la prueba no se ejecutaron completamente. Al probar los resultados de la prueba pueden ser inestables. La causa de lo anormal necesita ser identificada, corregida y vuelto a ejecutar antes de cualquier actividad de pruebas adicionales. Si las pruebas terminan anormalmente se continúa con la recuperación por pruebas detenidas.

3. Verificar los resultados de las pruebas

Los resultados de las pruebas deben verificarse para asegurar que son fiables, se deben registrar las fallas, advertencias o resultados inesperados que no fueron causados por influencias externas.

Las fallas más comunes de los procedimientos de pruebas que se ejecutan completamente y sus acciones correctivas se dan por:

- Pruebas de fallas de comprobación - esto ocurre cuando el resultado real y el resultado esperado no son iguales. Verificar que los métodos utilizados se enfocan en los elementos esenciales y en sus propiedades.
- Ventanas inesperadas - esto ocurre por varias razones. El más común es cuando una ventana no es la que debe estar activa o el número de ventanas desplegadas es mayor

que el esperado. Se debe verificar que el ambiente de pruebas ha sido inicializado como fue pensando para la ejecución.

- Ventanas perdidas - esta falla se da cuando se espera que una ventana esté disponible y no es así. Se debe verificar que el ambiente de pruebas ha sido inicializado como fue planeado para la ejecución. Verifique las ventanas reales perdidas.

Si el reporte de fallas es por errores identificados en los artefactos de pruebas o por problemas en el ambiente de pruebas debe tomarse la acción correctiva apropiada y volverse a ejecutar las pruebas.

Si los resultados de las pruebas indican que las fallas son genuinamente debidas al objetivo de pruebas, entonces la actividad de ejecución está completa.

4. Recuperación de las pruebas detenidas

Hay dos grandes tipos de pruebas detenidas:

- Errores fatales - el sistema falla por fracasos de la red, el hardware no responde, etc.
- Fallas de comando - específico al hacer pruebas automatizadas, esto es cuando una prueba no puede ejecutar un comando o una línea de código.

Ambos tipos de terminación anormal pueden exhibir los mismos síntomas:

- Muchas acciones inesperadas, ventanas o eventos que ocurren mientras las pruebas se están ejecutando.
- El ambiente de pruebas parece estar en un estado inmóvil.

Para hacer la recuperación de las pruebas detenidas, hacer lo siguiente:

- Determinar la causa real del problema.
- Corregir el problema.
- Restablecer el ambiente de pruebas.
- Reinicializar el ambiente de pruebas.
- Volver a ejecutar las pruebas.

Producir la unidad de distribución

Escribir las notas de liberación

Propósito

Describir los principales rasgos y cambios en la liberación. Las notas de liberación también deben describir cualquier error del sistema y las limitaciones al utilizar el producto.

Considerando que la factura de materiales proporciona una lista detallada de lo que constituye el producto y el plan de integración describe lo que esta contenido para cada

TESIS CON
FALLA DE ORIGEN

construcción, las notas de liberación proporcionan un resumen de lo que cambió desde la última versión. Las notas de liberación se adjuntan a la unidad de distribución.

La sugerencia es crear notas de liberación para la valoración al final de cada iteración. Sin embargo, las notas de liberación pueden actualizarse y mantenerse para cada construcción.

Desarrollar los artefactos de instalación

Propósito

Producir todo el software requerido para instalar y desinstalar el producto de manera rápida, fácil y segura sin afectar otras aplicaciones o características del sistema.

El desarrollo de los artefactos de instalación tienen que considerar cómo el producto será empaquetado, cómo se personalizará y cómo se instalará.

Un conjunto típico de artefactos de instalación incluye:

- Comandos de instalación
- Archivos de instalación
- Instrucciones de instalación

Deben desarrollarse artefactos de instalación con el mismo rigor y proceso que se sigue para desarrollar cualquier otro artefacto. Se pueden usar los requerimientos específicos para hacer los juegos de artefactos de instalación para cada plataforma.

El escenario de instalación más simple es donde el usuario individual instala el software en una sola computadora personal. Sin embargo, en organizaciones de desarrollo más grandes la responsabilidad es del administrador de red, las tareas de administración de software son globales. En este caso, hay gran beneficio al instalar y poner al día software rápida y fácilmente en todas las estaciones de trabajo desde un solo lugar. El precio de esta conveniencia se paga por la complejidad del software de instalación requerida para distribuir el software y personalizar las estaciones de trabajo desde el servidor.

El desarrollador puede querer usar algunos paquetes disponibles comercialmente que proporcionan plantillas y objetos de interfaces preconstruidas para hacer fácil de empaquetar el software para la instalación. Estos paquetes pueden ajustarse para desplegar imágenes para el producto y logotipos corporativos.

Generar la unidad de distribución

Propósito

Una unidad de distribución consiste de una construcción (una colección ejecutable de componentes), documentos (material de apoyo para el usuario final y notas de liberación) y artefactos de instalación. El propósito de esta actividad es crear una unidad de distribución que esté completa para ser descargada, instalada y sea ejecutada en un nodo como un grupo.

Todos los artefactos del proyecto se almacenan físicamente en el repositorio de proyectos (podría estar distribuido a en varias computadoras) y se organiza lógicamente de acuerdo a la estructura del producto. La unidad de distribución contiene todos los elementos entregables y éstos se listan en la factura de materiales.

En esta actividad, el administrador de configuración crea una copia de los elementos entregables, líneas base bajo el control de versiones en el repositorio del proyecto hacia los medios de comunicación necesarios para la distribución en el ambiente designado. Los medios de comunicación necesarios podrían ser un disco compacto o un sitio web donde el producto esté compactado y listo para ser descargado.

Administrar pruebas beta

Propósito

Las pruebas beta son una preliberación del producto dirigidas a una audiencia que probará el producto. Tienen dos propósitos, primero el producto se prueba en el mundo real y segundo se proporciona una vista previa de la próxima liberación. El administrador de distribución necesita administrar el programa para las pruebas beta del producto.

Pasos

1. Seleccionar los aspectos críticos del producto beta

Es una buena idea para una empresa guardar una base de datos sobre aspectos críticos de versiones beta y un archivo de retroalimentación. El administrador de distribución puede decidir sobre el perfil de las características de los aspectos críticos que se guardaran en la base de datos así como las personas.

2. Preparar y distribuir la unidad de distribución

El administrador de distribución necesita hacer la unidad de distribución (consiste de una construcción, el material de apoyo para el usuario final, las notas de liberación y los artefactos de instalación). La unidad de distribución puede enviarla o ponerla disponible en una página web. Los participantes en el programa del producto beta necesitarán contraseñas y licencias.

El administrador de distribución se asegura que cualquier usuario de los productos beta ha firmado y ha aceptado los términos del programa beta. El acuerdo beta debe cubrir la cláusula de terminación, duración de la revisión, la confidencialidad y no modificación o traslado del producto.

El administrador de distribución debe proporcionar un correo electrónico para la retroalimentación sobre las notas de liberación y los materiales de usuario final que acompañan al producto beta. Otro método para obtener retroalimentación es tener la capacidad de responder consultas en línea.

TESIS CON
FALLA DE ORIGEN

3. Resultados de revisiones y solicitud de cambios

El administrador de distribución recoge y revisa la retroalimentación y las solicitudes de cambio. Las solicitudes de cambio son parte del mecanismo de control de cambios. El control de cambios determina la importancia relativa de cada demanda y si se autoriza el cambio. El administrador de distribución debe preparar un resumen de evaluación de resultados e informar sobre la retroalimentación que se obtuvo del producto beta.

4. Proporcionar retroalimentación de los aspectos críticos

El administrador de distribución debe informar de las revisiones beta valoradas y dar una contestación personal a problemas particulares.

Empaquetar el producto

Lanzamiento industrial

Propósito

Producir en masa la versión del producto.

Pasos

1. Recoger los artefactos de acuerdo con la factura de materiales

La factura de materiales identifica todas las partes del producto o unidad de distribución. El administrador de distribución necesita asegurarse que el software ha sido probado y aceptado, además de que los elementos del producto están disponibles para la entrega en masa.

Todos los detalles de preparación para los elementos entregables son listados en el Plan de Distribución.

2. Entregar los artefactos a la organización industrial

Una vez que todas las partes del componente están en su lugar y se verificó que esta completo, pueden entregarse a la organización industrial para la producción de masa y duplicación.

Verificar el producto fabricado

Propósito

Asegurar que el producto fabricado está completo y es utilizable. Esta actividad a veces es llamada primera inspección del artículo. Sirve como una actividad de control de

calidad para asegurar que el producto tiene todos los atributos requeridos y los artefactos involucrados.

Pasos

1. Verificar el producto con la factura de materiales

El administrador de distribución necesita saber que el producto fabricado está completo. Es necesario hacer un chequeo e identificar los componentes del producto basándose en la factura.

2. Seguir las instrucciones de instalación del producto

Habiendo asegurado que el producto fabricado tiene todos los elementos requeridos, el administrador de distribución tiene que comprobar que el cliente puede cargar y puede ejecutar el software. Este paso es un chequeo para verificar la claridad de instrucciones de la instalación y qué trabajen bien los artefactos de instalación.

3. Verificar que el producto se puede utilizar

Habiendo instalado el software el administrador de distribución debe verificar las guías didácticas y debe probar el producto. Debe ver la ayuda del producto y la documentación para asegurar que todo está en orden.

Como una precaución se debe ver el reporte de errores, llamar la ayuda en línea, etc.

4. Enviar el producto a diferentes clientes

Asumiendo que el producto pasa las pruebas visuales y de usabilidad el administrador de distribución puede señalar la liberación del producto y hacerlo disponible para enviarlo a los clientes. El producto a enviar va más allá del alcance de un proceso de desarrollo de software, sin embargo, se supone que la empresa establece los cauces del mercadeo correctos y la infraestructura para la distribución del producto.

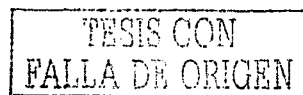
Crear la presentación del producto

Propósito

La presentación del producto se hace para publicar directamente como copia dura del producto que se empaquetó o una copia blanda en un sitio web. La presentación debe reforzar el producto, qué estándares propone y la creación de los mensajes apropiados para el consumidor.

Un artista gráfico debe ser consciente de los aspectos creativos y técnicos para la presentación del producto.

La creatividad de un artista gráfico decide la vista y el esquema de la presentación del producto y cómo aparecerá en la caja que debe tener en cuenta todos los pliegues, las



alas flexibles, el espesor y los colores que serán utilizados. En el caso de crear un sitio web, la preocupación del artista gráfico será el área de la imagen, la apelación visual y la navegación en el sitio del producto.

Proporcionar acceso a un sitio para la descarga

Propósito

asegurar que el producto está disponible para la compra y descarga a través de Internet.

Pasos

1. Agregar archivos del producto al servidor

Los clientes deben poder visualizar un sitio en internet con los archivos solicitados, deben poder comprar el producto a través del servidor. La implicación no sólo es que haya una presencia corporativa en la web sino que también haya seguridad en las transacciones de comercio electrónico.

El administrador de distribución necesita poner la unidad de distribución y cualquier otro producto con información pertinente sobre el servidor en un directorio de acceso predeterminado.

2. Habilitar el acceso del cliente al producto

El administrador de distribución es responsable de publicar y mantener el sitio web corporativo. El sitio debe incluir una página que describa el producto, debe permitir proporcionar acceso al producto y a los archivos relacionados.

El sitio web debe permitirle al cliente que haga compras en línea, transmitir el producto, mantener indicadores de apoyo del producto y respuestas a preguntas frecuentes.

3. Habilitar la retroalimentación del cliente y las capacidades de apoyo

De igual manera que se recibe retroalimentación de programas para pruebas beta, el sitio web del producto debe prepararse para solicitar retroalimentación del cliente.

Apéndice C

Flujo de trabajo pruebas de acuerdo al Proceso Unificado de Rational

El propósito de este flujo es asegurar que el sistema cumple con los requerimientos establecidos.

Actividades

Hacer el plan de pruebas

Propósito

Colectar y organizar la información para planear las pruebas y crear el plan de pruebas.

Identificar los requerimientos para las pruebas

Identificar los requerimientos para las pruebas es el comienzo de la actividad de planeación de las pruebas. Los requerimientos para las pruebas identifican lo que será probado junto con el alcance y el propósito del esfuerzo de las pruebas. Los requerimientos a probar también son utilizados para determinar el esfuerzo global de las pruebas.

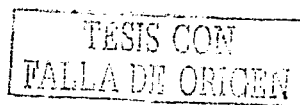
Los elementos que serán identificados como requerimientos para las pruebas deben ser verificables, es decir, deben ser medibles. Un requerimiento que no es verificable no es un requerimiento para hacer pruebas.

Consideraciones para identificar los requerimientos:

- Revisar todos los materiales.
- Indicar los requerimientos para las pruebas.

Revisar los materiales

Las fuentes más comunes de requerimientos incluye la lista de requerimientos, los casos de uso, el modelo de casos de uso, las realizaciones de casos de uso, especificaciones suplementarias, diseño de requerimientos, casos de negocio, entrevistas y la revisión de sistemas existentes.



Indicar los requerimientos para las pruebas

Se indica que requerimientos van a ser probados. Esto conduce a jerarquizar los requerimientos según su importancia.

El resultado de este paso es un reporte donde se identifican esos requerimientos que serán probados.

Evaluar riesgos

Consideraciones para evaluar riesgos:

- Identificar y justificar un factor de riesgo
- Identificar y justificar un factor del perfil operacional
- Identificar y justificar un factor de prioridad de prueba

Identificar y justificar un factor de riesgo

Los requerimientos más importantes para las pruebas son aquellos que reflejan los riesgos más altos.

Perspectivas de los riesgos:

- Efecto ---- Impacto o consecuencias de que falle un requerimiento.
- Causa ---- Razón por la que el requerimiento generó un resultado indeseable.
- Probabilidad --- Posibilidad de que falle el requerimiento.

Cada requerimiento a probar debe revisarse e identificar el factor de riesgo como alto, medio o bajo. A veces la evaluación del riesgo dos o más veces con diferentes perspectivas pueden producir un factor de riesgo diferente. En estas situaciones se usa el valor del factor de riesgo más alto. Debe hacerse una declaración del factor de riesgo que fue seleccionado para un requerimiento dado.

Identificar y justificar un factor del perfil operacional para las pruebas.

La mayoría de aplicaciones tiene funciones que se usan frecuentemente y otros que se usan esporádicamente. Por consiguiente, para probar una aplicación de manera aceptable, se debe asegurar que no sólo sean requerimientos de alto riesgo, sino también aquellos que se usan frecuentemente.

Identificar y justificar un factor de prioridad de prueba.

El factor de prioridad de prueba identifica la importancia relativa del requerimiento de la prueba y el orden en que se probará.

El factor de prioridad de prueba es identificado con los factores de riesgo, perfiles operacionales, obligaciones contractuales, otras necesidades o una combinación de todos.

éstos. Es importante considerar todos estos factores al identificar la prioridad de las pruebas.

Desarrollar las estrategias de las pruebas

El propósito de las estrategias de las pruebas es comunicar a todos cómo se realizará la comprobación y qué medidas se usarán para el cumplimiento y éxito. La estrategia no tiene que ser detallada pero debe ser una indicación de cómo se probará.

Consideraciones para establecer las estrategias:

- Identificar y describir como se probará
- Identificar el criterio para hacer las pruebas
- Identificar cualquier consideración especial

Identificar y describir las declaraciones

Se describe cómo se llevarán a cabo las pruebas. Las acciones a tomar mientras se está probando y cómo se verificarán los resultados, se debe proporcionar la información para que se pueda entender lo que se probará, aunque la profundidad de las pruebas no se conozca todavía, por ejemplo:

- Para cada caso de uso se identificarán los casos de prueba, ejecutando e incluyendo datos de entrada válidos e inválidos.
- Los procedimientos de prueba serán implementados para cada caso de uso.
- Los procedimientos de prueba serán implementados para simular cuentas del cliente por un periodo de tres meses. Los procedimientos deben incluir altas, modificaciones y bajas de las cuentas de los clientes.
- Los procedimientos de prueba serán implementados junto con los scripts de prueba para probar con cierto numero de usuarios virtuales, ejecutando las funciones A, B y C con diferentes entradas.

Identificar los criterios para las pruebas

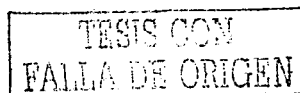
Los criterios para las pruebas deben ser declaraciones objetivas que indiquen los valores que determinan que las pruebas están completas y la calidad de la aplicación de las pruebas.

Consideraciones para identificar los criterios:

- Qué es lo que está probándose.
- Cuáles son las medidas a utilizar.
- Qué criterio está usándose para evaluar las medidas.

Muestra de los criterios de prueba, para los casos de uso de alta prioridad:

- Todos los casos y procedimientos de prueba fueron ejecutados.
- Se identificaron todos los defectos.



- Todos los casos de prueba y procedimientos se volvieron a ejecutar y no se identificó ningún nuevo defecto.

En el ejemplo anterior, lo que está probándose se describe por la declaración "para cada caso de uso de alta prioridad". Cómo se realizará la medición se indica con la declaración "todos los casos de uso y procedimientos de prueba han sido ejecutados". El criterio utilizado para la evaluación se incluye en las últimas dos declaraciones "Se identificaron todos los defectos" y "todos los casos de prueba y procedimientos de prueba se volvieron a ejecutar y no se identificó ningún nuevo defecto."

Identificar recursos

Una vez que se identificó lo que será probado y cómo, existe la necesidad de identificar quién hará las pruebas y lo que se necesita para apoyar estas actividades, identificando recursos como los siguientes:

- Recursos humanos (número de personas y sus habilidades)
- Ambiente de pruebas (hardware y software)
- Herramientas
- Datos

Identificar necesidades de recursos humanos

Para la mayoría de las pruebas se necesitarán recursos que puedan hacer lo siguiente:

- Administrar y planear las pruebas.
- Diseñar las pruebas y los datos.
- Implementar las pruebas y los datos.
- Ejecutar las pruebas y evaluar los resultados.
- Administrar y mantener los sistemas de prueba.

Identificar necesidades de recursos no humanos

Ambiente de pruebas (hardware y software), se recomiendan dos ambientes físicos diferentes (aunque no es necesario):

- El ambiente de aplicación, donde la administración de pruebas, diseño y actividades de implementación ocurren, y
- El ambiente de ejecución, un sistema de ejecución separado (normalmente un clon del sistema de producción) - donde todas las pruebas sean ejecutadas
- El software que será necesario para probar. El software mínimo necesario para hacer las pruebas de la aplicación, el cliente del sistema operativo, la red y el servidor del sistema operativo. Adicionalmente el software quizá necesite simular con precisión el ambiente de producción, este software podría incluir:
 - Interfaces a otros sistemas
 - Aplicaciones de escritorio
 - Aplicaciones que residan en los servidores de archivos y de red. Éstas son aplicaciones que, mientras no sean requeridos por la aplicación al hacer las pruebas residen en el ambiente mientras se ejecutan las pruebas.

Herramientas

Qué herramientas de software de prueba se usarán y por quién, así mismo, qué información o beneficios serán obtenidos por el uso de cada herramienta.

Datos

El software de pruebas cuenta con el uso de datos de entrada y salida esperados, deben identificarse estrategias para los datos de prueba:

- Colección o generación de datos para probar (entrada y salida).
- Arquitectura de la base de datos (aislamiento de influencias externas y métodos para devolver datos a su estado inicial después que hayan terminado las pruebas).

Hacer el calendario

Consideraciones para crear el calendario:

- Estimar el esfuerzo de las pruebas
- Generar el calendario de las pruebas

Estimar el esfuerzo de las pruebas

Consideraciones para asumir las estimaciones para las pruebas:

- La productividad y habilidad, nivel de conocimiento de los recursos humanos que trabajan en el proyecto (habilidad de usar herramientas de prueba)
- Los parámetros sobre la aplicación (como número de ventanas, componentes, entidades de datos, relaciones y porcentaje de reuso)
- Cobertura de las pruebas (la cobertura aceptable para que las pruebas sean implementadas y ejecutadas) usualmente muchos casos de prueba se requieren para la aceptación de los casos de uso o requerimientos

Generar el calendario de pruebas

Un calendario de pruebas del proyecto puede construirse del trabajo estimado y de la asignación de recursos. En el ambiente de desarrollo iterativo el calendario de pruebas de un proyecto debe estar separado por cada iteración. Todas las actividades de pruebas son repetidas en cada iteración.

En una iteración en particular las actividades de planeación y diseño de pruebas se dirigen a nuevas funciones del software. La implementación de las pruebas involucra la creación de nuevos casos de prueba para las nuevas funciones y los casos de prueba modificados para funciones que han cambiado. La ejecución de las pruebas y los pasos de evaluación validan nuevas funciones y realizan las pruebas de regresión para funciones existentes.

Las iteraciones tempranas introducen un número más grande de nuevas funciones y nuevas pruebas. Cuando el proceso de integración continúa el número de pruebas nuevas disminuye y un número creciente de pruebas de regresión necesita ser ejecutado para validar las funciones acumuladas. Por consiguiente, las iteraciones tempranas requieren

más trabajo en la planeación y diseño de pruebas, mientras que en iteraciones posteriores se enfocan hacia la ejecución y su evaluación.

Usando el esfuerzo estimado y los recursos asignados se establece el calendario. La tabla del ejemplo de abajo resume todas las actividades de las pruebas. Las estimaciones de trabajo se muestran como pautas para la cantidad de trabajo relativa para cada tarea.

Cuando se establece el calendario hay que asegurarse que es realista. Se puede desmoralizar con calendarios ambiciosos en que nadie tiene tiempo o energía para seguir y en el peor caso los resultados de las pruebas se llevan a cabo sin éxito.

Tabla general para el calendario de pruebas

Tareas	Trabajo relativo
Esfuerzo total	38d
Planear las pruebas	7d
Identificar pruebas del proyecto	1d
Definir las estrategias de pruebas	1d
Estimar el trabajo	1d
Identificar recursos	1d
Hacer el calendario de actividades de prueba	1d
Hacer el documento del plan de pruebas	2d
Especificar casos de prueba	5d
Determinar casos de prueba	5d
Diseñar las pruebas	7d
Analizar requerimientos de prueba	2d
Especificar procedimientos de prueba	3d
Especificar casos de prueba	1d
Revisar la cobertura de requerimientos de prueba	1d
Implementar las pruebas	12d
Establecer el ambiente de implementación de pruebas	1d
Registrar y rastrear los scripts prototipos	1d
Desarrollar los procedimientos de pruebas	5d
Revisar y poner a punto los procedimientos de prueba	1d
Modificar procedimientos de prueba	2d
Establecer el juego de datos externos	1d
Volver a probar y poner a punto los procedimientos	1d
Ejecutar pruebas de sistema	6d
Preparar las pruebas	1d
Ejecutar las pruebas	2d
Verificar resultados esperados	1d
Estudiar resultados inesperados	1d
Registrar los defectos	1d
Evaluar las pruebas	1d
Revisar los registros de las pruebas	0.25d
Evaluar la cobertura de los casos de prueba	0.25d
Evaluar defectos	0.25d
Determinar si se completaron las pruebas	0.25d

Generar el plan de pruebas

Consideraciones para generar el plan de pruebas:

- Revisar / refinar los materiales existentes
- Identificar pruebas entregables
- Generar el plan de pruebas

Revisar / refinar los materiales existentes

Antes de generar el plan de pruebas, hay que hacer una revisión de toda la información del proyecto existente para asegurar que el plan de pruebas contiene la información más actual y exacta. Si es necesario, la información relacionada con las pruebas como requerimientos, estrategias, recursos, etc. debe revisarse para reflejar cualquier cambio.

Identificar pruebas entregables

El propósito de la sección de pruebas de entregables es identificar y definir cómo los artefactos de prueba se crearán, se mantendrán y se harán disponibles a otros miembros del equipo. Estos artefactos incluyen:

- Modelo de pruebas
- Casos de prueba
- Procedimientos de prueba
- Scripts de prueba
- Solicitudes de cambio

Generar el plan de pruebas

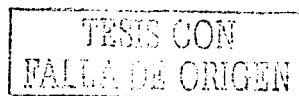
Una vez que se genera el plan de pruebas se debe distribuirse a:

- Todos los involucrados en roles de pruebas
- Representante de los desarrolladores
- Representante de afectados
- Representante del cliente
- Representante del usuario final

Diseñar las pruebas

Propósito

Identificar un conjunto de casos de prueba comprobables para cada construcción e identificar los procedimientos de prueba.



Realizar el análisis de carga de trabajo

El análisis de la carga de trabajo es realizado para generar un documento que ayude a ejecutar las pruebas. Las entradas primarias de un documento de análisis de carga de trabajo incluyen:

- Plan de desarrollo de software
- Modelo de casos de uso
- Modelo de diseño
- Especificaciones suplementales

El análisis de carga de trabajo incluye lo siguiente:

1. Clarificar los objetivos de ejecución de pruebas y casos de uso.
2. Identificar los casos de uso a ejecutar en el modelo.
3. Identificar los actores y características del actor a simular en la ejecución de las pruebas.
4. Identificar la carga de trabajo a simular en la ejecución de las pruebas (en términos de número de actores / clases de actores y perfiles de actores).
5. Determinar las medidas de desempeño y criterios.
6. Revisar los casos de uso e identificar la frecuencia de ejecución.
7. Seleccionar los casos de uso invocados frecuentemente y aquéllos que generan la mayor carga en el sistema.
8. Generar los casos de prueba para cada caso de uso.
9. Identificar los puntos críticos a medir para cada caso de prueba.

Identificar y describir los casos de prueba

Para cada requerimiento a probar:

- Analizar el flujo de trabajo de la aplicación
- Identificar y describir casos de prueba
- Identificar datos para los casos de prueba

Analizar los flujos de trabajo de la aplicación

El propósito es identificar, describir acciones y los pasos del actor al usar el sistema. Las descripciones de los procedimientos de prueba son utilizadas para identificar y describir los casos de prueba necesarios para probar la aplicación.

Nota: Estas descripciones de procedimiento de prueba deben ser de alto nivel, es decir, las acciones deben describirse genéricamente tanto como sea posible sin referencias específicas a componentes reales u objetos.

Para cada caso de uso o requerimiento:

- Revisar el flujo de eventos del caso de uso, o
- Describir los acciones o pasos que el actor debe hacer al utilizar interactivamente el sistema

Identificar y describir los casos de prueba

El propósito es establecer qué casos de prueba son apropiados para las pruebas de cada requerimiento.

Nota: Si las pruebas de una versión previa ya se han llevado a cabo, ya existen casos de prueba. Estos casos de prueba deben revisarse para ser utilizados y diseñados como pruebas de regresión. Los casos de prueba de regresión deben ser incluidos en la iteración actual y combinarse con los nuevos casos de prueba con un nuevo comportamiento.

Las entradas primarias para identificar casos de prueba son:

- Los casos de uso que en algún punto recorren algún destino de las pruebas como el sistema, subsistema o componente.
- El modelo de diseño.
- Cualquier requerimiento técnico o suplementario.
- Mapa de la aplicación del destino de las pruebas (por ejemplo, un script de prueba generada por una herramienta automatizada).

Describir los casos de prueba:

- La condición de la prueba.
- El caso de uso, escenario del caso de uso o el requerimiento técnico de casos de prueba derivados.
- El resultado esperado, condición o valores de datos.

Se obtiene una matriz de casos de prueba que identifican las condiciones, los objetos, datos o influencias en el sistema que crea la condición que será probada y el resultado esperado.

Identificar los datos del caso de prueba

Usando la matriz revisar los casos de prueba e identificar los valores reales que apoyan los casos de prueba.

- Los valores de datos utilizados como entrada
- Los valores de datos para los resultados esperados
- Los datos que apoyan el caso de prueba sin utilizarlos como entrada o salida de un caso de prueba específico

TESIS CON
FALLA DE ORIGEN

Identificar y estructurar los procedimientos de prueba

- Revisar el flujo de la aplicación
- Desarrollar el modelo de pruebas
- Estructurar los procedimientos de prueba

Revisar el flujo de la aplicación

Se revisa el flujo de la aplicación y los procedimientos de prueba para determinar si hay cambios en el caso de uso.

Si se utiliza una herramienta automática para generar scripts de prueba se debe revisar el flujo de la aplicación para asegurar que la jerarquía de objetos de control de interfaz es correcta respecto a los casos de uso.

- Revisar el flujo de eventos del caso de uso
- Revisar la descripción de los procedimientos de prueba
- Recorrer los pasos que el actor debe tomar al probar el sistema
- Revisar la aplicación

Desarrollar el modelo de pruebas

El modelo de pruebas se usa para comunicar lo que será probado, cómo se probará y cómo se llevará a cabo.

- Identificar la relación o secuencia del procedimiento de prueba con otros procedimientos de prueba
- Identificar la condición final o estado final para el procedimiento de prueba
- Indicar los casos de prueba a ejecutar por el procedimiento de prueba

Consideraciones para desarrollar el modelo de pruebas:

- Muchos casos de prueba son variantes de otros, lo que podría significar que pueden ser satisfechos por el mismo procedimiento de prueba.
- Muchos casos de prueba pueden requerir un comportamiento de sobreposición al ser ejecutados. Para poder reutilizar la implementación de ese comportamiento se debe elegir la estructura de los procedimientos de prueba para ser utilizados en varios casos de prueba.
- Muchos procedimientos de prueba incluyen acciones o pasos que son comunes a muchos casos de prueba e incluso a otros procedimientos de prueba. En estos casos, debe determinarse si un procedimiento de prueba estructurado se realizará por separado (para esos pasos comunes) mientras los pasos de prueba específicos permanecen en un procedimiento de prueba estructurado por separado.
- Al usar una herramienta automática para generar scripts de prueba revisar la aplicación y los scripts de prueba generadas para asegurar que el modelo de pruebas es correcto.

Estructurar los procedimientos de prueba

La estructuración apropiada de los procedimientos de prueba incluyen revisión para incluir la siguiente información:

- ✔ Estructuración: crear condiciones para los casos de prueba y los datos necesarios.
- ✔ Condición inicial, estado o acción para la estructura del procedimiento de prueba.
- ✔ Instrucciones para la ejecución, los pasos o acciones detalladas tomadas por el probador al implementar y ejecutar las pruebas (al grado de declarar el objeto o componente).
- ✔ Los valores de datos a introducir.
- ✔ Resultado esperado (condición, datos, o referencias al caso de prueba) para cada acción o paso.
- ✔ Evaluación de resultados: el método y pasos utilizados para analizar el resultado real obtenido comparándolos con los resultados esperados.
- ✔ Condición final, estado o acción para la estructura del procedimiento de prueba.

Nota: Cuando se estructura un procedimiento de prueba se debe maximizar la reutilización y minimizar el mantenimiento.

Revisar y evaluar la cobertura de las pruebas

Identificar medidas para la cobertura de las pruebas

Hay dos métodos para determinar la cobertura de las pruebas:

- ✔ Cobertura basada en requerimientos.
- ✔ Cobertura basada en el código.

La cobertura basada en requerimientos utiliza casos de uso, requerimientos, flujos de casos de uso o condiciones de prueba como la medida del total de elementos probados, además de que se usarán para hacer el plan de pruebas.

La cobertura basada en el código utiliza el código generado como el elemento para conocer si las pruebas están completas y como medida de que el código ha sido ejecutado durante las pruebas (como las líneas de código ejecutadas o el número de caminos recorridos). Este tipo de medida sólo puede llevarse a cabo después de que el código ha sido generado.

Identificar el método a utilizar y mencionar cómo se realizará la medición, cómo deben interpretarse los datos y cómo se usará la métrica durante el proceso.

Generar y distribuir reportes sobre la cobertura de las pruebas

Identificar en el plan de pruebas el calendario de cuándo se realizarán los reportes de la cobertura de las pruebas y cuándo se distribuirán. Estos informes deben distribuirse a:

- ✔ Todos los roles involucrados en las pruebas
- ✔ Representante de los desarrolladores
- ✔ Representante del inversionista



Implementar las pruebas

Registrar y generar los scripts de prueba

- Generar los scripts de prueba
- Probar y depurar los scripts de prueba
- Revisar y evaluar la cobertura de las pruebas

Generar los scripts de prueba

Consideraciones para generar scripts de prueba:

1. Maximizar la reutilización de los scripts de prueba
2. Minimizar el mantenimiento de los scripts de prueba
3. Utilizar scripts existentes cuando sea factible
4. Utilizar las herramientas de prueba para crear scripts de prueba en lugar de programarlos cuando sea factible
5. Aplicar objetos GUI y acciones de los métodos más estables (como el nombre del objeto o el uso del botón del ratón)

Consideraciones para depurar los scripts de prueba:

1. Configurar del ambiente de pruebas (si es necesario)
2. Reiniciar el ambiente
3. Ejecutar los scripts de prueba
4. Evaluar resultados
5. Determinar las siguientes acciones apropiadas:
 - a. Resultados esperados: No hay acciones
 - b. Resultados inesperados: Determinar la causa del problema y solución

Revisar y evaluar la cobertura de las pruebas

En la realización de creación y generación de scripts de prueba los reportes de cobertura de pruebas serán generados para verificar que los scripts han logrado la cobertura deseada.

Identificar la funcionalidad de las pruebas específicas en los modelos de diseño e implementación

Identificar la funcionalidad de las pruebas específicas que debe ser incluida en el modelo de diseño y en el modelo de implementación. La definición de funcionalidad de pruebas específicas se lleva a cabo durante las pruebas de integración donde existe la necesidad de mantener controladores para los componentes o sistemas que no están incluidos todavía.

Hay dos estilos:

- Controladores que son simplemente esqueletos sin funcionalidad capaces de recibir un valor específico o regresar un valor predefinido.
- Controladores que son más inteligentes y pueden simular comportamientos más complejos.

Se debe usar el segundo estilo prudentemente porque éste utiliza más recursos. Un equilibrio entre valor agregado y el esfuerzo necesario para implementar y probar los controladores necesarios.

Describir los resultados de los requerimientos de prueba específicos en los modelos de diseño e implementación.

Establecer los conjuntos de datos externos

Generar y mantener conjuntos de datos externos

1. Revisar el modelo de pruebas, casos de prueba y procedimientos de prueba
2. Generar conjuntos de datos utilizando métodos y herramientas
3. Modificar los scripts de prueba para utilizar los conjuntos de datos

Diseñar paquetes y clases de prueba

Un paquete es un mecanismo para organizar elementos que tienen objetivos comunes en grupos, pueden anidarse paquetes dentro de otros paquetes.

Identificar paquetes y clases para las pruebas

Basado en las entradas, el diseñador de pruebas identifica y especifica las clases de pruebas y paquetes específicos en el diseño de modelo

Un controlador de un diseño de una clase tiene los mismos métodos que la clase original, pero no tiene ninguna conducta definida por los métodos, se usa para mantener la entrada o regresar un valor predefinido

El controlador de un paquete a diseñar contiene clases simuladas para las clases que forman la interfaz pública del paquete original.

Diseñar la interfaz de la herramienta de pruebas automatizada

Identificar el comportamiento de la herramienta de automatización de pruebas para comunicarse con el objetivo de las pruebas de manera eficaz. Identificar y describir las clases del diseño y los paquetes.

Diseñar el comportamiento del procedimiento de pruebas

Para automatizar los procedimientos de prueba cuando no hay ninguna herramienta de automatización identificar las clases de diseño y paquetes. Utilizar los casos de prueba y los casos de uso que se derivan de una entrada.

Implementar los componentes y subsistemas de prueba

Implementar y generar las pruebas de unidad

Identificar y diseñar las clases y paquetes que proporcionarán la funcionalidad de las pruebas.

Implementar y hacer la interfaz de pruebas de unidad para la herramienta de pruebas automatizada

Identificar la interfaz necesaria para la integración de una herramienta de prueba automatizada con una funcionalidad de prueba específica.

Implementar y generar el comportamiento del procedimiento de las pruebas de unidad

Automatizar los procedimientos de prueba cuando no hay ninguna herramienta de pruebas automatizada.

Ejecutar las pruebas de integración

Propósito

Ejecutar pruebas y registrar los resultados.

Ejecutar los procedimientos de prueba

- Configurar el ambiente de pruebas para asegurar que todos los componentes necesarios se han implementado y se encuentran en el ambiente.
- Inicializar el ambiente de pruebas para asegurar que todos los componentes están en el estado inicial correcto para el comienzo de las pruebas.
- Ejecutar los procedimientos de prueba.

Pruebas automatizadas: Se ejecutan scripts de prueba generados durante la actividad de implementación de pruebas.

Pruebas manuales: Se usan los procedimientos de prueba desarrollados durante la actividad de diseño de pruebas.

Evaluar la ejecución de las pruebas

La ejecución de pruebas que terminan con alguna condición:

Normal: Todos los procedimientos de prueba se ejecutaron como se requería y están completas. Si las pruebas terminaron normalmente se continúa con verificación de los resultados de prueba:

Anormal: Los procedimientos de prueba no se ejecutaron completamente o como se deseaba. Al probar rangos de datos limite los resultados de las pruebas pueden ser inestables. La causa de la terminación anormal necesita ser identificada, corregida volverse a ejecutar antes de cualquier actividad de pruebas adicionales.

Si las pruebas terminan anormalmente continúa con la recuperación de pruebas fallidas.

Verificar los resultados de las pruebas

Los resultados deben revisarse para asegurar que son fiables e informar de los fracasos, advertencias o resultados inesperados no causados por influencias externas, como puede ser una configuración inadecuada.

Los fracasos más comunes son:

- Fallas de verificación de pruebas - esto ocurre cuando el resultado real y el resultado esperado no concuerdan. Verificar que el método utilizado se enfoca únicamente a los elementos esenciales y propiedades.
- Ventanas inesperadas - esto ocurre por varias razones. El más común es cuando una ventana llama a otra que no es esperada o el número de ventanas es mayor que el esperado. Asegurar que el ambiente de pruebas ha sido configurado o reiniciado como fue especificado para una ejecución apropiada.
- Ventanas perdidas - esta falla se produce cuando se espera que una ventana esté disponible y no es así. Asegurar que el ambiente de pruebas ha sido configurado o reiniciado como fue especificado para una ejecución apropiada. Verificar que las ventanas perdidas reales son ajustadas.
- Si los fracasos reportados se deben a errores identificados en los artefactos de prueba o debido a problemas con el ambiente de pruebas se toma la acción correctiva apropiada y se vuelven a ejecutar las pruebas.

Recuperar el sistema después de pruebas fallidas

Hay dos tipos de pruebas fallidas:

- Errores fatales - el sistema falla (fracasos de la red, el hardware colisiona, etc.)
- Fallas en comandos de scripts de prueba - específico en pruebas automatizadas, esto es cuando el script de prueba no puede ejecutar un comando.

TESIS CON
FALLA DE ORIGEN

Ambos tipos de terminación anormal a probar pueden exhibir los mismos síntomas:

- Acciones inesperadas, ventanas o eventos que ocurren mientras el script de prueba se está ejecutando.
- El ambiente de pruebas parece insensible o en un estado indeseable.

Para recuperarse de pruebas fallidas hacer lo siguiente:

- Determinar la causa del problema
- Corregir el problema
- Reconfigurar el ambiente de pruebas
- Reiniciar el ambiente de pruebas
- Reejecutar las pruebas

Ejecutar las pruebas de sistema

Propósito

Ejecutar pruebas y registrar los resultados.

Ejecutar los procedimientos de prueba

- Configurar el ambiente de pruebas para asegurar que todos los componentes necesarios se han implementado y se encuentran en el ambiente.
- Inicializar el ambiente de pruebas para asegurar que todos los componentes están en el estado inicial correcto para el comienzo de las pruebas.
- Ejecutar los procedimientos de prueba.

Pruebas automatizadas: Se ejecutan scripts de prueba generados durante la actividad de implementación de pruebas.

Pruebas manuales: Se usan los procedimientos de prueba desarrollados durante la actividad de diseño de pruebas.

Evaluar la ejecución de las pruebas

La ejecución de pruebas que terminan con alguna condición:

Normal: Todos los procedimientos de prueba se ejecutaron como se requería y están completas. Si las pruebas terminaron normalmente se continúa con verificación de los resultados de prueba:

Anormal: Los procedimientos de prueba no se ejecutaron completamente o como se deseaba. Al probar rangos de datos límite los resultados de las pruebas pueden ser inestables. La causa de la terminación anormal necesita ser identificada, corregida volverse a ejecutar antes de cualquier actividad de pruebas adicionales.

Si las pruebas terminan anormalmente continúa con la recuperación de pruebas fallidas.

Verificar los resultados de las pruebas

Los resultados deben revisarse para asegurar que son fiables e informar de los fracasos, advertencias o resultados inesperados no causados por influencias externas, como puede ser una configuración inadecuada.

Los fracasos más comunes son:

- Fallas de verificación de pruebas - esto ocurre cuando el resultado real y el resultado esperado no concuerdan. Verificar que el método utilizado se enfoca únicamente a los elementos esenciales y propiedades.
- Ventanas inesperadas - esto ocurre por varias razones. El más común es cuando una ventana llama a otra que no es esperada o el número de ventanas es mayor que el esperado. Asegurar que el ambiente de pruebas ha sido configurado o reiniciado como fue especificado para una ejecución apropiada.
- Ventanas perdidas - esta falla se produce cuando se espera que una ventana esté disponible y no es así. Asegurar que el ambiente de pruebas ha sido configurado o reiniciado como fue especificado para una ejecución apropiada. Verificar que las ventanas perdidas reales son ajustadas.
- Si los fracasos reportados se deben a errores identificados en los artefactos de prueba o debido a problemas con el ambiente de pruebas se toma la acción correctiva apropiada y se vuelven a ejecutar las pruebas.

Recuperar el sistema después de pruebas fallidas

Hay dos tipos de pruebas fallidas:

- Errores fatales - el sistema falla (fracasos de la red, el hardware colisiona, etc.)
- Fallas en comandos de scripts de prueba - específico en pruebas automatizadas, esto es cuando el script de prueba no puede ejecutar un comando.

Ambos tipos de terminación anormal a probar pueden exhibir los mismos síntomas:

- Acciones inesperadas, ventanas o eventos que ocurren mientras el script de prueba se está ejecutando.
- El ambiente de pruebas parece insensible o en un estado indeseable.

Para recuperarse de pruebas fallidas hacer lo siguiente:

- Determinar la causa del problema
- Corregir el problema
- Reconfigurar el ambiente de pruebas
- Reiniciar el ambiente de pruebas
- Reejecutar las pruebas

TESIS CON
FALLA DE ORIGEN



Evaluar las pruebas

Propósito

Evaluar los resultados de las pruebas y hacer solicitudes de cambio, calcular y entregar medidas de las pruebas y generar el reporte de evaluación.

Analizar los resultados de las pruebas y hacer solicitudes de cambio

Los resultados de la prueba se analizan para identificar detalles con respecto a diferencias entre los resultados obtenidos y los resultados esperados. Las diferencias indican defectos potenciales en las pruebas y deben registrarse en la solicitud de cambios para tomar las acciones correctivas.

Revisar cada falla reportada e identificar la información necesaria para hacer una solicitud de cambios en el seguimiento del sistema. La información introducida debe ser exacta, apropiada y conocida por todos, la solicitud de cambios debe hacerse de acuerdo a los estándares y guías establecidos.

Evaluar la cobertura de las pruebas basadas en los requerimientos

Para evaluar la cobertura de las pruebas basadas en requerimientos es necesario revisar los resultados de las pruebas para determinar:

- La proporción entre cuántas pruebas basadas en requerimientos (casos de prueba) han sido ejecutados en esta iteración y el número total de pruebas.
- La proporción de casos de prueba con éxito.

Evaluar la cobertura de las pruebas basadas en el código

Para evaluar la cobertura de las pruebas basado en el código es necesario revisar los resultados y determinar:

- La proporción entre el código que se ha ejecutado durante las pruebas (líneas o declaraciones) en esta iteración y el código total del destino de las pruebas.

El objetivo es asegurar que 100% del código para esta iteración se han ejecutado con éxito.

Analizar defectos

Para analizar los defectos es necesario revisar y analizar las medidas seleccionadas como parte de las estrategias de análisis de defectos. Las medidas de defectos más comunes utilizadas incluyen:

- Densidad de defectos - El número de defectos se muestra como una función de los atributos del defecto (como estado o severidad).
- Tendencia de defectos - La cuenta del defecto se muestra como una función sobre el tiempo.

- Antigüedad del defecto - Un reporte de densidad de defecto especial en el que los números de defectos se muestran como una función de longitud de tiempo donde el defecto permanece en un cierto estado (abierto, nuevo, esperando a ser probado, etc.) Comparar las medidas de esta iteración con los resultados del análisis de iteraciones anteriores para conseguir una vista de tendencias de defectos.

Es aconsejable mostrar los resultados en forma de diagrama.

Determinar si se han completado las pruebas y si el criterio de éxito se logró

Revisar la estrategia de pruebas definida en el plan de pruebas. Debe presentar los criterios de prueba en términos de cobertura de pruebas o evaluación de defectos. Examinar los resultados de las pruebas, defectos, análisis de los defectos y determina si el criterio se ha cumplido.

Si el criterio no se ha cumplido hay varias alternativas:

Recolectar información adicional:

- Producir diferentes reportes, tales como reportes de densidad de defectos
- Investigar el proceso para determinar si las condiciones inesperadas contribuyeron a la desviación de criterios de prueba y reevalúe el criterio basado en este nuevo conocimiento

Sugerir un calendario de pruebas adicionales:

- Implementar nuevas pruebas para agregar profundidad a los casos de prueba
- Implementar nuevas pruebas para agregar mayor cobertura de pruebas

Modificar el criterio de pruebas:

- Revisar y evaluar el riesgo de cambiar el criterio de pruebas
- Identificar un subconjunto de software que satisfaga el criterio de pruebas y decidir si puede ser utilizado.

Generar el reporte de evaluación de pruebas

Generar el resumen de evaluación de prueba. Esto se logra reuniendo la información anterior en un reporte y distribuyéndolo a los roles apropiados para la revisión.

TESIS CON
FALLA DE ORIGEN

Apéndice D

Ejemplo de llenado de los artefactos del Flujo de Trabajo Ambiente

TESIS COM
FALLA DE ORIGEN

Herramientas a comprar

Siglas del sistema PSIMex Ciclo 1
Responsable JASC Fecha

Paso: Identificar las herramientas a comprar

Jefe del Departamento de Informática
Presente

Por este conducto le solicito a usted sea tan amable de realizar las gestiones necesarias para comprar las herramientas que se listan a continuación.

Nombre de la herramienta	Versión	Sistema operativo donde opera
Inno Setup Installer	4	Windows 9x,2000,XP,ME

Esta herramienta se utilizará para realizar la instalación del sistema de Publicaciones Sobre la Independencia de México a los usuarios finales.

Sin más por el momento me despido de usted enviándole un cordial saludo.

Atentamente

Nombre
Especialista en herramientas

Instituto de Investigaciones Bibliográficas
Departamento de Informática
Sección de desarrollo de Software



Configuración de herramientas

Siglas del sistema PSIMex Ciclo 1
 Responsable JASC Fecha 11/05/03

Paso: Indicar la configuración de las herramientas (Revisar el "Estándar de herramientas")

Nombre de la herramienta	Configuración	Verificado
Servidor CIRCE:Windows NT 3.5	<ol style="list-style-type: none"> 1. Crear un grupo de trabajo psimex 2. Crear el directorio de trabajo psimex. 3. Crear el subdirectorio bd dentro del directorio psimex donde se colocará la base de datos. 	Si
Servidor CIRCE:Windows NT 3.5	<ol style="list-style-type: none"> 1. Las cuentas a crear enseguida deben tener acceso al directorio de trabajo psimex. 2. Crear la cuenta "jasc" con la contraseña "jasc999" 3. Crear la cuenta "fab" con la contraseña "fab999" 4. Crear la cuenta "pau" con la contraseña "pau999" 5. Crear la cuenta "tar" con la contraseña "tar999" 6. Crear la cuenta "cha" con la contraseña "cha999" 	Si
Windows 2000 computadoras para desarrollo	1. Colocar la biblioteca de funciones Isis32.dll versión 4 en el directorio windows\system32	Si

Vo. Bo.

 Nombre
 Administrador del proyecto

TESIS CON
 FALLA DE ORIGEN

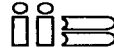
Apéndice E

Ejemplo de llenado de los artefactos del Flujo de Trabajo Distribución

TESIS CON
FALLA DE ORIGEN

Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software



TRABAJA CON
FALLA DE ORIGEN

Plan de distribución

Siglas del sistema PSIMex Ciclo 1
Responsable JASC Fecha _____

Paso: Definir acciones que debe hacer el usuario responsable

Recursos necesarios

Hardware:

Solicitar una computadora Pentium III a 1 Ghz, 30 GB DD, 128 RAM.

Software:

Red:

Solicitar un nodo de red para el cubículo 105 del Edificio B.

Otros:

Paso: Definir herramientas que acompañan al sistema

Sistema operativo:

Manejador de base de datos: Winisis 1.4

Otros:

Paso: Definir recursos materiales para la distribución

# Discos compactos		# Discos flexibles	1 caja de 10 discos
# Cartuchos para impresora de inyección de tinta	1 HP51626A negro	# Toner para impresora laser	
# Paquetes de hojas blancas	1		
Otros (especificar)			

Paso: Definir migración de datos

* Usar un manejador de base de datos

Nombre: _____ Versión: _____

* Conversión de formato

Nombre del programa: _____

* Desde archivos

Nombre del programa: _____

* Otro: No hay necesidad de hacer migración de datos

Paso: Listar casos de uso a entregar

Nombre del módulo o caso de uso	Nombre del módulo o caso de uso
Captura	Entrar
	Guardar
	Modificar
	Nuevo
	Salir

Paso: Listar documentación a liberar

Documentación a liberar

Manual de configuración

Si No

Manual de instalación

Manual de usuario

Manual técnico

Otro (especificar)

TESIS CON
FALLA DE ORIGEN

**TESIS CON
FALLA DE ORIGEN**

Definir forma de capacitar al usuario (Seleccionar solo una, poner una "X")

- Capacitación a un usuario responsable
- Capacitación a todos los usuarios
- Capacitación por subgrupos de usuarios
- No hay capacitación

Donde será la capacitación (Seleccionar solo una, poner una "X")

- Auditorio del IIB
- Departamento de Informática
- Sala de usos múltiples del segundo piso
- Sala de usos múltiples del cuarto piso
- Cubículo 302 Edificio B

Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software



Invitación de pruebas formales

Siglas del sistema PSIMex Ciclo 1
Responsable JASC Fecha _____

Paso: Preparar invitación a los interesados de las pruebas.

Nombre del usuario responsable Fabiola Monroy Valverde

Por este conducto informamos que el día 25 de mayo a las 10:00 horas se realizaran las pruebas de liberación del sistema Publicaciones Sobre la Independencia de México en el Departamento de Informática sobre los siguientes módulos.

Lista de casos de uso o módulos

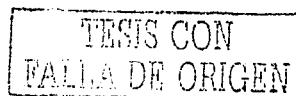
Nombre del caso de uso o módulo	Nombre del caso de uso o módulo
Captura	Entrar
	Guardar
	Modificar
	Nuevo
	Salir

Nombres de los usuarios invitados

Tarsicio García Díaz
Fabiola Monroy Valverde
Paulo Cu Mena

Atentamente

José Antonio Salazar Carmona
Administrador de distribución



TESIS CON
FALLA DE ORIGEN



Instituto de Investigaciones Bibliográficas
Departamento de Informática
Sección de desarrollo de Software

Aceptación de pruebas en desarrollo

Siglas del sistema PSIMex Ciclo 1
Responsable JASC Fecha _____

Por este conducto informamos que el sistema Publicaciones Sobre la Independencia de México cumple con los requerimientos establecidos en los siguientes casos de uso:

Lista de casos de uso o módulos liberados

Nombre del caso de uso o módulo	Nombre del caso de uso o módulo
Captura	Entrar
	Guardar
	Modificar
	Nuevo
	Salir

Atentamente

Vo. Bo.

José Antonio Salazar Carmona
Administrador de distribución

Fabiola Monroy Valverde
Usuario responsable

Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software



Aceptación de pruebas en sitio

Siglas del sistema PSIMex Ciclo 1
Responsable JASC Fecha _____

Por este conducto informamos que el sistema Publicaciones Sobre la Independencia de México cumple con los requerimientos establecidos en los siguientes casos de uso:

Lista de casos de uso o módulos liberados

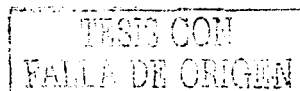
Nombre del caso de uso o módulo	Nombre del caso de uso o módulo
Captura	Entrar
	Guardar
	Modificar
	Nuevo
	Salir

Vo. Bo.

Fabiola Monroy Valverde
Usuario responsable

Recibió

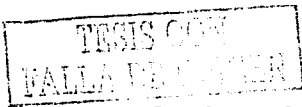
Ana Yuri Ramírez Molina
Administrador del proyecto



Apéndice F

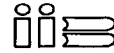
Ejemplo de llenado de los artefactos del Flujo de Trabajo Pruebas

TESIS CON
FALLA DE ORIGEN



Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software



Plan de pruebas

Siglas del sistema PSIMex Ciclo 1
Responsable JASC Fecha _____

Paso: Identificar casos de uso a probar

Utilizar la siguiente tabla para nombrar los casos de prueba

Estrategia de prueba	Prefijo para nombrar los casos de prueba
Pruebas de unidad (caja blanca)	PU
Pruebas de función (caja negra)	PF
Pruebas de ciclo de negocio	PCN
Pruebas de base de datos	PBD
Pruebas de interfaz de usuario	PIU
Pruebas de instalación	PI
Pruebas de configuración	PC

Caso de uso o módulo	Caso de prueba
Captura	PBD_Captura
	PIU_Captura
Captura:Entrar	PIU_Entrar
Captura:Guardar	PU_Guardar
	PF_Guardar
	PBD_Guardar
Captura:Modificar	PU_Modificar
	PF_Modificar
	PBD_Modificar
Captura:Nuevo	PU_Nuevo
	PF_Nuevo
	PBD_Nuevo
Captura:Salir	PIU_Salir

Nombre del Sistema	Caso de prueba
PSIMex	PI PSIMex
	PC PSIMex

TESIS CON
FALLA DE CENCEN

Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software



Caso de prueba

Siglas del sistema PSIMex Ciclo 1
Responsable JASC Fecha _____

Paso: Identificar los casos de prueba

Nombre del caso de uso o módulo a probar Captura

Paso: Describir los casos de prueba

Nombre del caso de prueba PI PSIMex

Precondiciones
Cerrar todas las aplicaciones de windows.

Postcondiciones
El sistema de PSIMex debe quedar instalado en la computadora para utilizar el módulo de captura.

Procedimiento de prueba

#	Acción a realizar	Entrada de datos	Resultado esperado	Resultado de la prueba
1	Ejecutar el programa instalar.exe	No se necesitan datos		
2	Presionar el botón aceptar			
3	Presionar el botón iniciar instalación			
4	Presionar el botón ok de finalización de instalación		Debe quedar un icono llamado PSIMex en el menú programas de windows	Correcto

Fecha de ejecución _____

Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software



Caso de prueba

Siglas del sistema PSIMex Ciclo 1
Responsable JASC Fecha _____

Paso: Identificar los casos de prueba

Nombre del caso de uso o módulo a probar Captura

Paso: Describir los casos de prueba

Nombre del caso de prueba PIU_Captura

Precondiciones
Comenzar en la casilla de contraseña

Postcondiciones
Terminar en la casilla de notas

Procedimiento de prueba

#	Acción a realizar	Entrada de datos	Resultado esperado	Resultado de la prueba
1	Presionar la tecla de tabulador para recorrer todos las casillas de captura	Se hace la prueba sin datos en las casillas	La secuencia del tabulador es Autor, título de libro, edición, lugar de publicación, editor, año, páginas, detalles físicos, dimensiones, otros autores, temas, localización, datos del autor, juicio crítico, resumen y notas.	Correcto
2	Presionar la tecla enter para recorrer las casillas de captura	Se hace la prueba sin datos en las casillas	Las casillas que deben aceptar la tecla enter son: Autor, título de libro, edición, lugar de publicación, editor, año, páginas, detalles físicos,	Correcto

TESIS CON
FALLA DE ORIGEN

**TESIS CON
 TALLA DE ORIGEN**

			dimensiones, otros autores, temas, localización	
3	Presionar la tela enter sobre las casillas: Datos del autor, Juicio crítico, resumen y notas	Se hace la prueba sin datos en las casillas	Se debe agregar una línea en blanco cada vez que se pulsa la tecla enter	Correcto

Fecha de ejecución _____

Bibliografía

[Booch] Booch, Grady., Rumbaugh, James., Jacobson, Ivar. "El lenguaje Unificado de Modelado". Addison Wesley. (1999)

[Curtis] Curtis, Bill., Kellner, Marc I., Over, Jim. "Process Modeling. Communications of the ACM", 1992, vol. 35(9), 75-90.

[Flores] Flores Ríos, Brenda L., Marínez García, Ana I., "A Computer Aided Process Engineering Tool for the Study of Organizational Process". Sociedad Mexicana de Ciencia de la Computación: Memorias del tercer encuentro Internacional de Ciencias de la Computación ENC 01, 2001, 939-948.

[Huff] Huff, Karen E. "Software Process Modeling. Software Process". Edited by Fuggetta and Wolf, 1996, pág. 1-22.

[Jacobson] Jacobson, Ivar., Booch, Grady., Rumbaugh, James. "El Proceso Unificado de Desarrollo de Software". Addison Wesley. (2000)

[Kruchten] Kruchten, Philippe. "The Rational Unified Process an Introduction". Second Edition, Addison Wesley. (2000)

[Ramírez] Ramírez, A. Y. "La implementación y administración del proyecto, cambios y configuración del Proceso Unificado adaptados a un caso práctico". Tesis de maestría en Ciencias de la Computación UNAM.

[Rumbaugh] Rumbaugh, James., Jacobson, Ivar., Booch, Grady., "El Lenguaje Unificado de Modelado. Manual de Referencia". Addison Wesley. (2000)

[RUP] Página Web local del Proceso Unificado de Rational instalado en el laboratorio del Postgrado de Ciencia e Ingeniería de la Computación.

[Salazar] Salazar, J. A. "Pruebas, Ambiente y Distribución basados en el Proceso Unificado (un caso práctico)". Tesis de maestría en Ciencias de la Computación UNAM.

[Sánchez] Sánchez, M. G. "Adaptación del Proceso Unificado en los flujos de trabajo de requerimientos, análisis y diseño: caso práctico". Tesis de maestría en Ciencias de la Computación UNAM.

TESIS CON
FALLA DE ORIGEN