

Tema 1: Introducción a la Ingeniería en Software

Software

Software: En general se define al software como un conjunto de programas y la documentación que lo acompaña.

Existen tres tipos básicos de software:

- System Software
- Utilitarios
- Software de Aplicación

Algunos problemas con el desarrollo de software

- La versión final del producto no satisface la necesidad del cliente.
- No es fácil extenderlo y/o adaptarlo. Agregar mas funcionalidad en otra versión es casi una misión imposible.
- Mala documentación.
- Mala calidad.
- Mas tiempo y costos que los presupuestados.

Tema 2: Software Configuration Management (SCM)

Cambios en el software

Tienen su origen en:

- Cambios en el software y nuevos requerimientos
- Soporte de cambios de productos asociados
- Reorganización de las prioridades de la empresa por crecimiento
- Cambios en el presupuesto
- Defectos encontrados a corregir
- Oportunidades de mejora

Definición

Una disciplina que aplica dirección y monitoreo administrativo y técnico a:

- Identificar y documentar las características funcionales y técnicas de los ítems de configuración
- Controlar los cambios de esas características
- Registrar y reportar los cambios y su estado de implementación y verificar correspondencia con los requerimientos.

Elementos de SCM

1. Identificación de Ítems de Configuración

Se trata de establecer estándares de documentación y un esquema de identificación de documentos.

2. Control de cambios

Consiste en la evaluación y registro de todos los cambios que se hagan de la configuración de software.

3. Auditorías de Configuración

Sirven, junto a las revisiones técnicas formales, para garantizar que el cambio se ha implementado correctamente.

4. Generación de informes

Identificación de ítems de configuración

Son todos aquellos artefactos que forman parte del producto o participan de la gestión del proyecto que puedan sufrir cambios o necesitan ser compartidos entre los miembros del equipo

y sobre los cuales necesitamos conocer su estado y evolución. Pueden ser requerimientos, documentos de diseño, código fuente, código ejecutable, etc.

Control de cambios

Tiene su origen en un **Requerimiento de cambio** sobre uno o muchos ítems de configuración que se encuentran en una línea base.

Es un procedimiento formal que involucra muchos actores y una evaluación del impacto del cambio.

Repositorio

Es un depósito de información conteniendo todos los ítems de configuración. Mantiene la historia de cada CI con sus atributos y relaciones (metainformación). Es utilizado para hacer **evaluaciones de impacto** de los cambios propuestos. Pueden ser una o varias bases de datos. Posee herramienta de automatización, integración y generación de informes.

Repositorios Centralizados

[TODO: Insertar gráfico de PDF]

Un servidor contiene todos los archivos con sus versiones. Los administradores tienen mayor control sobre el repositorio. **Falla el servidor y estamos al H O R N O.**

Repositorios Descentralizados

[TODO: Insertar gráfico de PDF]

Cada cliente tiene una copia exactamente igual del repositorio completo. Si un servidor falla solo es cuestión de “copiar y pegar”. Posibilita otros workflows no disponibles en el modelo centralizado.

Auditorías de Configuración

- **Auditoría física de configuración (PCA):** Asegura que lo que está indicado para cada SCI en la línea base o en la actualización se ha alcanzado realmente.

- **Auditoría funcional de configuración (FCA):** Evaluación independiente de los productos de software, verificando que la funcionalidad y performance reales de cada ítem de configuración sean consistentes con la especificación de requerimientos.

Características de las Auditorías

- Las auditorías cuestan tiempo y dinero.
- Deben realizarse desde la primeras etapas de desarrollo. Su postergación a etapas posteriores puede llegar a hacer fracasar el proyecto.
- Suministra visibilidad y rastreabilidad del ciclo de vida del producto de software.

Registro e Informe de Estado

- Se ocupa de mantener los registros de la evolución del sistema
- Maneja mucha información y salidas por lo que se suele implementar dentro de procesos automáticos
- Incluye reportes de rastreabilidad de todos los cambios realizados a las líneas base durante el ciclo de vida.

La configuración

Un conjunto de ítems de configuración con su correspondiente versión en un momento determinado.

La Línea Base

- Una configuración que ha sido revisada formalmente y sobre la que se ha llegado a un acuerdo
- Sirve como base para desarrollos posteriores y puede cambiarse sólo a través de un procedimiento formal de control de cambios
- Permiten ir atrás en el tiempo y reproducir el entorno de desarrollo en un momento dado del proyecto

Se utilizan “etiquetas” para marcar las baseline. NO confundir con la versión del producto x.y.z.

Plan de Gestión de Configuración

Qué debería incluir el plan?

- Reglas de nombrado de los CI
- Herramientas a utilizar para SCM

- Roles e integrantes del Comité
- Procedimiento formal de cambios
- Plantillas de formularios
- Procesos de Auditoría

SCM en entornos Agile

- **Individuos e interacciones sobre procesos y herramientas**

Los procesos y herramientas de SCM deben adaptarse al equipo y su forma de trabajo y no al revés.

- **Software funcionando sobre documentación extensiva**

Es posible minimizar la dependencia del equipo de trabajo en procedimientos documentados automatizando tareas propias de la gestión de SCM con herramientas.

- **Colaboración con el cliente sobre negociación contractual**

El uso de herramientas de SCM apropiadas puede proveer mayor visibilidad a los stakeholders del estado del proyecto y mejorar la comunicación.

- **Respuesta ante el cambio sobre seguir un plan**

El uso de las estructuras y políticas de SCM adecuadas debe facilitar y propiciar el cambio en vez de prevenirlo.

El mapa de los patrones

[TODO: Insertar gráfico del PDF]

Antipatrones

- **Merge-Paranoia:** Evitar la fusión a toda costa, por lo general a causa de un temor a las consecuencias.
- **Merge-Mania:** Gastar demasiado tiempo la fusión de configuración, en lugar de su desarrollo.
- **Big Bang Merge:** Aplazar la fusión hasta el final de las actividades de desarrollo y de intentar fusionar todas las ramas simultáneamente.
- **Development Freeze:** Detener desarrollo mientras que ejecutan ramificaciones, fusiones, o la construcción de nueva línea base.
- **Never-Ending Merge:** Fusiones continuas, porque siempre queda algo por integrar.
- **Wrong-Way Merge:** Fusionar una versión de componentes con una versión obsoleta.
- **Branch Mania:** Generar ramificaciones sin una versión aparente.
- **Cascading Branches:** Generar ramificaciones, pero nunca actualizar la línea base.
- **Berlin Wall:** Las ramificaciones dividen al equipo de desarrollo, en lugar de dividir su trabajo

Testing

Asegurar la calidad vs controlar la calidad

- Una vez definidos los requerimientos de calidad, tengo que tener en cuenta que:
 - La calidad no puede “inyectarse” al final
 - La calidad del producto depende de tareas realizadas durante todo el proceso
 - Detectar errores en forma temprana ahorra esfuerzos, tiempo, recursos
 - La calidad no solamente abarca aspectos del producto sino también del proceso y como estos se pueden mejorar, que a su vez evita defectos recurrentes.
 - El testing NO puede asegurar ni calidad en el software ni software de calidad

Definición de prueba de software

Visión más apropiada del Testing:

- Proceso destructivo de tratar de encontrar defectos (cuya presencia se asume) en el código.
- Se debe ir con una actitud negativa para demostrar que algo es incorrecto.
- Testing exitoso -> Es el que encuentra defectos
- Mundialmente: 30 a 50% del costo de un software confiable

Testing Unitario

- Se prueba cada componente tras su realización/construcción
- Solo se prueban componentes individuales
- Cada componente es probado de forma independiente
- Se produce con acceso al código bajo pruebas y con el apoyo del entorno de desarrollo, tales como un framework de pruebas unitarias o herramientas de depuración.
- Los errores se suelen reparar tan pronto como se encuentran, sin constancia oficial de incidentes.

Testing de Integración

- Test orientado a verificar que las partes de un sistema que funcionan bien aisladamente, también lo hacen en conjunto
- Cualquier estrategia de prueba de versión o de integración debe ser incremental, para lo que existen dos esquemas principales:
 - Integración de arriba hacia abajo

- Integración de abajo hacia arriba
- Lo ideal es una combinación de ambos esquemas
- Tener en cuenta que los módulos críticos deben ser probados lo más tempranamente posible.
- Los puntos clave del test de integración son simples:
 - Conectar de a poco las partes más complejas
 - Minimizar la necesidad de programas auxiliares

Testing de sistema

- Es la prueba realizada cuando una aplicación está funcionando como un todo.
- Trata de determinar si el sistema en su globalidad opera satisfactoriamente.
- El entorno de prueba debe corresponder al entorno de producción tanto como sea posible para reducir al mínimo el riesgo de incidentes debidos al ambiente específicamente y que no se encontraron en las pruebas
- Deben investigar tanto requerimiento funcionales y no funcionales del sistema

Testing de aceptación

- Es la prueba realizada por el usuario para determinar si la aplicación se ajusta a sus necesidades
- La meta en las pruebas de aceptación es el de establecer confianza en el sistema, las partes del sistema o las características específicas y no funcionales del sistema.
- Encontrar defectos no es el foco principales en las pruebas de aceptación.
- Comprende tanto la prueba realizada por el usuario en ambiente de laboratorio, como la prueba en ambientes de trabajo reales.

Proceso del testing

Planificación

- La planificación de las pruebas es la actividad de verificar que se entienden las metas y los objetivos del cliente, las partes interesadas, el proyecto y los riesgos de las pruebas que se pretende abordar.
- Construcción del test plan:
 - Riesgos y objetivos del testing
 - Estrategia de testing
 - Recursos
 - Criterio de aceptación
- Controlar:
 - Revisar los resultados del testing
 - Test coverage y criterio de aceptación.
 - Tomar decisiones

Identificación

- Revisión de la base de pruebas
- Verificación de las especificaciones para el software bajo pruebas
- Evaluar la testeabilidad de los requerimientos y el sistema
- Identificar los datos necesarios
- Diseño y priorización de los casos de las pruebas
- Diseño del entorno de prueba

Ejecución

- Desarrollar y dar prioridad a nuestros casos de prueba
- Crear los datos de prueba
- Automatizar lo que sea necesario
- Creación de conjuntos de pruebas de los casos de prueba para la ejecución de la prueba eficientemente.
- Implementar y verificar el ambiente
- Ejecutar los casos de prueba
- Registrar el resultado de la ejecución de pruebas y registrar la identidad y las versiones del software en las herramientas de pruebas
- Comparar los resultados reales con los resultados esperados.

Evaluación

- Evaluar los criterios de aceptación
- Reporte de los resultados de las pruebas para los stakeholders
- Recolección de la información de las actividades de pruebas completadas para consolidar
- Verificación de los entregables y que los defectos hayan sido corregidos
- Evaluación de cómo resultaron las actividades de testing y se analizan las lecciones aprendidas