

Clase: No.8

Asignatura: Gestión de Proyecto de software.

UNIDAD VI: Gestión de Cambio.

Tema: Planificación de la gestión de configuraciones.

Gestión del Cambio

Gestión de Versiones y Entregas.

Objetivos:

Que los Estudiantes puedan:

Comprender por qué es importante la gestión de la configuración del software.

Manejar las cuatro actividades de la gestión de cambio: Gestión de conversiones y entregas, construcción de sistemas.

GESTION DEL CAMBIO EN INGENIERÍA DE SOFTWARE

El cambio es inherente al software computacional y genera confusión entre los ingenieros de software involucrados en un proyecto, dicha confusión surge cuando los cambios no se analizan antes de realizarlos, no se registran antes de implementarlos, no se reporta a quienes deben saberlo o no se mantiene un control sobre ellos.

Esto es la gestión de la configuración: "El arte de coordinar el desarrollo de software para minimizar la confusión", GCS o GC.

La GC es una actividad que se desarrolla durante todo el proceso de desarrollo ya que no sabemos en que momento se originara un cambio, las actividad en este proceso se desarrollan para:

1. Identificar el Cambio
2. Controlar el Cambio
3. Garantizar que el cambio se realizara de manera adecuada
4. Reportar los cambios a todos los interesados

¿Quien es el encargado de la GC?

Todos los involucrados en el desarrollo de un proyecto deberían en alguna medida participar en la GC aun que hay empresas con mas recursos que tienen personas especializadas para este fin.

¿Por qué es importante?

Si el cambio no se controla en un proyecto de software, este tiene al caos a su alcance aun cuando hay una gran cantidad de personas involucradas y en posibles partes distintas, tanto así que un desarrollo de software sin control puede llegar a entregar productos de muy mala calidad por lo cual es una práctica sólida de ingeniería de software

¿Cuáles son los Pasos?

Primero se deben identificar los productos de trabajo, segundo establecer mecanismos para el control de versiones y cambio, tercero auditar el proceso para asegurarse que la calidad se mantiene en el cambio y que los interesados reciben la información requerida

¿Cuál es el producto obtenido?

Un plan de GC

¿Cómo estamos seguros que se ha hecho bien?

Cuando cualquier producto puede explicarse, seguirse y controlarse y los cambios pueden seguirse y analizarse y todos los interesados están enterados de un cambio que se ha hecho.

La primera ley de la ingeniería de sistemas {BER80} afirma:

No importa donde se encuentra el ciclo de vida del sistema, el sistema cambiará y el deseo de cambiarlo persistirá durante todo el ciclo de vida

Existen cuatro fuentes fundamentales del cambio

- a. Nuevas condiciones en el negocio
- b. Nuevas necesidades del cliente
- c. Reorganización o crecimiento del negocio
- d. Restricciones presupuestales

Elementos de un sistema de GC

1. Elementos de Componentes: herramienta que permite el acceso y gestión de cada elemento de GC
ejemplo: Base de datos
2. Elementos de proceso: serie de procedimientos y tareas que definen un enfoque eficaz con el cual gestionar el cambio
3. Elementos de construcción: Conjunto de herramientas que automatizan la construcción del software al asegurar que se ensamblado un conjunto de componentes válidos de software
4. Elementos humanos. que el equipo utilice las herramientas y procesos para GC

Etapas del proceso GC

1. Identificación: se deben nombrar cada uno de los elementos que intervienen mediante un enfoque orientado a objetos
2. Control de la versión: combina procedimientos y herramientas para gestionar diferentes versiones de objetos que se crean durante el proceso del software
3. Control de cambio. se deben crear procesos para generar cambios al sistema, donde se debe evaluar el impacto y otros aspectos.

4. Auditoria de la configuración

Ayuda a asegurar que el cambio se ha realizado con propiedad, abordando las siguientes preguntas

¿Se ha realizado el cambio específico?, ¿Se han incorporado modificaciones adicionales?

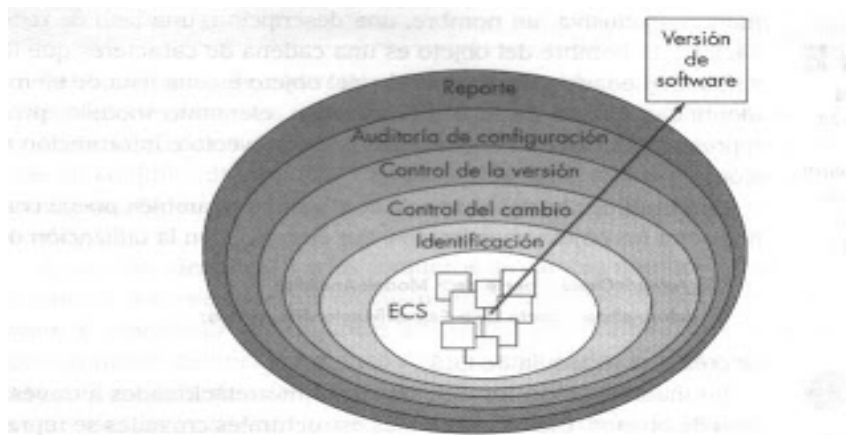
¿Se ha realizado una revisión técnica formal para evaluar la corrección técnica?

¿Se ha seguido el proceso de software? ¿Se han aplicado debidamente los estándares de ingeniería de software? ¿Se han regido los procesos de GC ?

5. Reporte, informe de estado

se resuelven las preguntas

¿Qué ocurrió? ¿Quién lo hizo? ¿Cuándo ocurrió? ¿Qué otra cosa será afectada?



Gestión de Configuración y Control de Cambios

La Gestión de Configuración y Control de Cambios es la disciplina que se encarga de

- Identificar los elementos del proyecto que deben estar bajo configuración
- Restringir los cambios a dichos elementos
- Auditar los cambios a estos elementos
- Definir y gestionar la configuración de estos elementos

Los métodos, procesos y herramientas utilizados para proveer este ambiente de configuración son parte esencial del proceso de desarrollo de software.

La Gestión de Configuración y Control de Cambios es esencial al momento de tener control sobre todos los elementos generados por los integrantes del equipo de proyecto. Este control ayuda a eliminar la posibilidad de confusiones que pueden resultar de alto costo para el proyecto y asegurar que no existan inconsistencias en el sistema desarrollado, generadas por:

- Actualizaciones simultáneas
Cuando varios integrantes del equipo trabajan sobre un mismo elemento al mismo tiempo.
- Problemas en la notificación de cambios

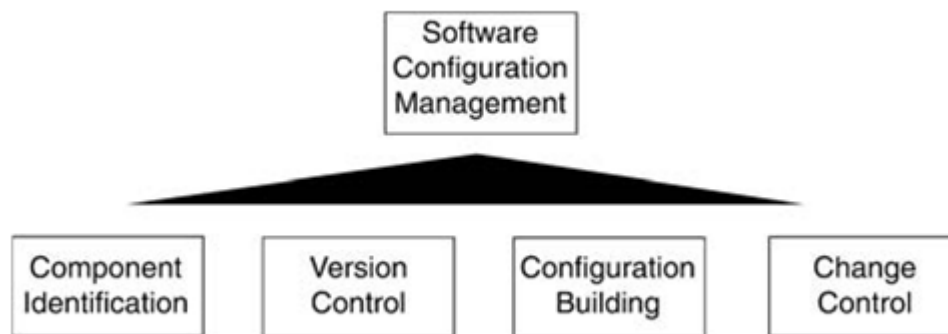
Cuando un problema fue resuelto para algún elemento que es compartido por varios desarrolladores y alguno de ellos no fue notificado de dicho cambio.

- Múltiples versiones

Usualmente se tienen varias versiones del producto en desarrollo, por ejemplo una versión de desarrollo, y otra de test, y se quiere que cuando haya un cambio en una éste se vea reflejado en las demás versiones.

Algunos de los beneficios que se obtienen de la realización de una buena Gestión de Configuración y Control de Cambios son:

- Brindar apoyo a los métodos de desarrollo de software.
- Mantener la integridad del producto.
- Asegurar la completitud y correctitud de los elementos bajo configuración.
- Proveer un ambiente estable y controlado de trabajo.
- Restringir y controlar los cambios que se realizan.
- Proveer mecanismos de rastreo de porqué, cuando y quién realizó un cambio.



Relación con otras Disciplinas

La disciplina Gestión de Configuración y Control de Cambios se relaciona con todas las disciplinas en la medida que evalúe los procedimientos seguidos para la realización de cambios y realice el seguimiento y auditorías sobre los elementos bajo configuración.

Control de cambios

El control de cambios es el proceso mediante el cual la modificación del software se propone, se evalúa, se aprueba o se rechaza, se programan y se le da seguimiento. Su fundamento básico es un proceso de control de cambios, un estado de los componentes, presentación de informes, y un proceso de auditoría. El control de cambios en el software es un proceso de decisión utilizado en el control de los cambios realizados al software. Algunos cambios propuestos son aceptados e implementados durante este proceso. Otros son rechazados o pospuestos, y no se aplican. El control de cambios también se proporciona para el análisis de impacto para determinar las dependencias.

La modificación de una configuración tiene al menos cuatro elementos: una solicitud de cambio, un análisis del impacto del cambio, un conjunto de modificaciones y adiciones de nuevos componentes, y un método fiable para la instalación de las modificaciones como una nueva línea de base, todo esto debe ser registrado en el documento de solicitud de cambio.

Un cambio a menudo implica modificaciones a los componentes de software.

Por lo tanto, un sistema de almacenamiento que proporciona varias versiones de un solo archivo no suele ser suficiente. Una técnica es necesaria para identificar el conjunto de modificaciones como un solo cambio. Esto a menudo se llama el almacenamiento delta. Cada componente de software tiene un ciclo de vida de desarrollo. Un ciclo de vida consiste en estados y las transiciones permitidas entre los estados. Cuando un componente del software se cambia, siempre debe ser revisado y almacenado hasta que una nueva versión sea creada. La autoridad revisora debe aprobar o rechazar las modificaciones realizadas al componente del software. Una librería de software cuenta con todos los componentes de software tan pronto como estos se almacenan y también actúa como un repositorio de componentes homologados.

Uno de los componentes derivados está ligado a su origen y tiene el mismo estatus que su fuente. Además, una configuración no puede tener un estado más completo que cualquiera de sus componentes, porque no tiene sentido para revisar una configuración cuando algunos de los componentes asociados no están almacenados o registrados.

Todos los componentes sometidos a control por software de gestión de configuración y almacenados en una biblioteca de configuración de software, incluidos los productos de trabajo tales como datos de negocio y modelos de procesos, los grupos de arquitectura, diseño de unidades, pruebas del software de aplicación, la reutilización del software, y el software de prueba especial

Cuando un componente de software se va a modificar, este es chequeado en el repositorio o librería en un espacio de trabajo privado. Se desarrolla a través de muchos estados, que se encuentran temporalmente fuera del ámbito de control de gestión de la configuración.

Cuando un cambio se ha completado, el componente se registró en la biblioteca y se convierte en una versión de software nuevo componente. El componente anterior la versión también se conserva.

El plan de aseguramiento de la calidad

El plan de aseguramiento de la calidad del software (SQA) es un esbozo de las medidas de calidad para garantizar niveles de calidad dentro de un esfuerzo de desarrollo de software. El plan se utiliza como referencia para comparar los niveles reales de calidad durante el desarrollo con los niveles de calidad previstos. Si los niveles de calidad no están dentro de los niveles de calidad previstos, la gerencia debe responder de manera adecuada como se documentó en el plan.

El plan constituye el marco y las directrices para el desarrollo de código comprensible y fácil de mantener. Estos ingredientes ayudan a garantizar la cualidad apreciada en un proyecto de software. Un plan de SQA también proporciona los procedimientos para garantizar que la calidad del software se produce o mantiene durante el desarrollo del software. Estos procedimientos afectan a la planificación, a el diseño, la codificación, las pruebas, la documentación, el almacenamiento y mantenimiento de los programas de informática.

Debe ser organizado de esta manera porque el plan garantiza la calidad del software en lugar de describir los procedimientos específicos para el desarrollo y mantenimiento del mismo.

Pasos para desarrollar y aplicar un Plan de Aseguramiento de Calidad de Software

Paso 1. Documento del Plan

El plan de aseguramiento (garantía) de la calidad del software debe incluir las siguientes secciones.

- **Sección Objetivo** - Esta sección delinea el propósito específico y ámbito particular del plan de SQA. Se debe indicar el nombre (s) de la elementos de software que abarca el plan de SQA y el uso previsto de los software. Afirma la parte del ciclo de vida del software cubierto por el plan de SQA para cada artículo del software especificado.
- **Sección de Documento de Referencia** - Esta sección incluye una lista completa de los documentos referenciados en cualquier lugar del plan de SQA.
- **Sección de Gestión** - En esta sección se describe la organización del proyecto su estructura, tareas y responsabilidades.

- **Sección de Documentación** - En esta sección se identifica la documentación que rigen el desarrollo, verificación validación, uso y mantenimiento del software. También afirma que los documentos son para comprobar su adecuación. Esto incluye los criterios y la identificación de las revisiones, auditorías que la adaptación de cada documento sea confirmado.
- **Sección de Estándares, prácticas, convenios, y mediciones** - En esta sección se identifica las normas, prácticas, convenciones y métricas que serán aplicadas, y también indica cómo el cumplimiento de estos elementos serán monitoreado de forma segura.
- **Sección de Revisiones e Inspecciones** - En esta sección se definen las técnicas de revisiones de gestión, tutoriales, y las inspecciones a realizar. También indica cómo los exámenes, tutoriales, y las inspecciones se pueden lograr incluyendo las actividades de seguimiento y aprobaciones.
- **Sección de la gestión de configuración de software** - Esta sección es tratada en detalle en el plan de la gestión de configuración del proyecto de configuración de software.
- **Sección de Problema de Información y Acción Correctiva** - Esta sección se trata en detalle el plan del proyecto de gestión de configuración del software.
- **Sección de la metodología, técnicas y herramientas** – En esta sección se identifican las herramientas especiales de software, técnicas y metodologías que apoyan SQA, los estados, sus propósitos, y describe su uso.
- **Sección de control de Código** - En esta sección se definen los métodos y las instalaciones utilizadas para mantener, almacenar, proteger, controlar y documentar las versiones del software identificado durante todas las fases de desarrollo. Esto puede ser implementado en conjunto con un programa de ordenador biblioteca y / o que puedan establecerse como parte de la configuración de software plan de manejo.
- **Sección de Control de los medios de comunicación** - Esta sección establece los métodos e instalaciones que se utilizarán para identificar los medios de comunicación para cada producto y equipo la documentación requerida para almacenar los medios de comunicación, incluyendo la copia y el proceso de restauración, y protege el programa de computadora física los medios de comunicación del acceso no autorizado o daño accidental o deterioro durante todas las fases de desarrollo. Esto puede ser proporcionado por la configuración de software de gestión del plan.
- **Sección de control del proveedor** - Esta sección establece las disposiciones para asegurar que el software proporcionado por los proveedores cumple establecido los requisitos. Además, debe indicar los métodos que se utilizan para asegurar que el proveedor de software recibe suficientes y los requisitos completos. Para previamente desarrollado un software, en esta sección establece los métodos que se utilizan para asegurar la idoneidad de los productos para su uso con los elementos de software cubierto por el plan de SQA. {Para el software a desarrollar, el proveedor estará obligado a preparar e implementar un plan de SQA de conformidad con esta norma. En esta sección también se indicarán los métodos que se emplearán para asegurar que los desarrolladores cumplan con los requisitos de esta norma.

- **Sección de retención, Colección y mantenimiento de Documentos** - En esta sección identifica la documentación SQA que deben conservarse. Afirmar los métodos y facilidades para montar, proteger y mantener esta documentación, y designar el período de retención. La puesta en práctica del plan de SQA implica las aprobaciones necesarias para el plan, así como el desarrollo de un plan para su ejecución. La posterior evaluación del plan de SQA se llevará a cabo como resultado de su ejecución.
- **Sección de Metodología de pruebas** - En esta sección se define el enfoque de prueba, técnicas y herramientas automatizadas que se utilizarán.

Paso 2. Obtener la aceptación de la Gerencia

La participación de la gerencia es necesaria para la implementación exitosa de un plan de SQA. La administración es responsable tanto para garantizar la calidad de un proyecto de software como de proporcionar los recursos necesarios para el desarrollo del mismo.

El nivel de compromiso de la gerencia necesarias para la aplicación de un plan de SQA depende del alcance del proyecto. Si un proyecto abarca las fronteras de la organización, la aprobación se debe obtener de todas las zonas afectadas. Una vez se obtenga la aprobación, el plan de SQA se coloca debajo del control de configuración.

En el proceso de aprobación de la gerencia, la gerencia renuncia a la administración y control del plan para la mejora de la calidad del software (SQA), el control del plan de aseguramiento de la calidad del software a menudo se deja al equipo de desarrolladores. La calidad es deseable, pero la gerencia puede expresar preocupación por los costos de un plan formal de SQA. El personal debe ser consciente de que la gerencia ve el programa como un medio de garantizar la calidad del software, y no como un fin en sí mismo.

Para hacer frente a problemas la gerencia debe tener un estimado oficial de los costos del ciclo de vida de desarrollo de los proyectos ejecutados con o sin plan de calidad. En general, la aplicación de un plan formal de SQA hace económica y da sentido a la gestión.

Paso 3. Obtener un desarrollo aceptable

Debido a que el personal de desarrollo de software y mantenimiento son los principales usuarios de un plan de SQA, su aprobación y su cooperación en la aplicación del plan es esencial. Los miembros del equipo del proyecto de software deben cumplir el plan de SQA, todo el mundo debe aceptarlo y seguirlo.

No hay un plan de SQA que se aplique con éxito sin la participación de los miembros del equipo de software y sus directivos en el desarrollo del plan. Debido a que los equipos de proyecto en general, tienen sólo unos pocos miembros, todos los miembros del equipo deberían participar activamente en la redacción del plan SQA. Cuando los proyectos son muy largos (es decir, que abarca las divisiones enteras o departamentos), los representantes de los subgrupos del proyecto deben proporcionar información. La retroalimentación constante de los representantes de los miembros del equipo ayuda al plan a ganar aceptación.

Paso 4. Plan de implementación del plan de SQA

El proceso de planificación, formulación y elaboración de un plan de SQA requiere personal y los recursos de procesamiento de textos. La persona responsable de la aplicación de un plan de SQA debe tener acceso a estos recursos. Además, la asignación de recursos requiere la aprobación de la gerencia y, en consecuencia, su apoyo.

Para facilitar la asignación de recursos, la gerencia debe ser consciente de los riesgos del proyecto que pueda impedir el proceso de aplicación (por ejemplo, la limitada disponibilidad de personal o de equipo). Un calendario para la redacción, revisión y aprobación del plan de SQA debe ser desarrollado.

Paso 5. Ejecutar el Plan de SQA

El proceso real de la ejecución de un plan de SQA por el equipo de desarrollo y mantenimiento de software consiste en determinar los puntos necesarios para el control de auditoría de éste. La función de auditoría debe ser programada durante la fase de ejecución del producto de software para que el control inadecuado del proyecto de software no afecte el plan de SQA. Los Puntos de Auditoría debe ocurrir ya sea de manera periódica durante el desarrollo o en momentos específicos del proyecto (Por ejemplo, en las revisiones importantes o cuando una parte del proyecto se entrega).

Identificación de los componentes

Una configuración básica de gestión de la actividad del software es la identificación de los componentes de software que componen una entrega en cada punto de su desarrollo. La gestión de configuración del Software proporciona directrices para identificar y nombrar las líneas de base de software, componentes de software y configuraciones del software. Los componentes de software pasan por una serie de cambios. Con el fin de gestionar el proceso de desarrollo, es necesario establecer métodos y normas, nombre para la identificación exclusiva de cada revisión. Una forma sencilla de nombre de las revisiones de componente consiste en utilizar una serie de cifras discretas. El primer entero podría referirse al número de un componente de software de desbloqueo exterior. El segundo entero podría representar el desarrollo interno de software número de versión. La transición de número de versión 2,9 a 3,1 indicaría que un nuevo versión externa 3 se ha producido. El componente de software número de versión se incrementa automáticamente cuando el componente se verifica en el software biblioteca. Además de los niveles de clasificación podría utilizarse también cuando sea necesario, como la fecha de una nueva versión. Una configuración de software es una colección de elementos de software que componen una función de negocio principales. Un ejemplo de una configuración es el conjunto de módulos del programa para un sistema de orden. La identificación de una configuración es bastante similar a la identificación de los componentes individuales de software. Configuraciones se pueden tener una secuencia de versiones. Cada configuración debe ser nombrada de una manera que lo distingue de los demás. Cada versión de configuración debe ser diferenciada de otras versiones. La identificación de una configuración debe

También incluyen su estado de aprobación y una descripción de cómo la configuración fue construido.

Una técnica simple para identificar una configuración para almacenar todos sus programas componentes en una sola biblioteca o repositorio. La lista de todos los componentes también puede ser documentada.

El control de versiones

Como las aplicaciones evolucionan con el tiempo, son creadas muchas versiones diferentes del software en ese transcurso, y es necesario que haya un proceso organizado de gestionar los cambios en los componentes de software y sus relaciones. Por lo tanto es un requisito el soporte para el desarrollo y mantenimiento de componentes de software en paralelo.

Los programas informáticos van cambiando a medida que evolucionan a través de una sucesión de estados temporales llamadas versiones, es necesario que exista una gestión de control de instalaciones de versiones de software en forma de biblioteca. El control de versiones ofrece la trazabilidad o un historial de cambio realizado en el software, incluyendo quién hizo? Qué?, por qué? y cuándo?. Dentro del ciclo de vida del software, los componentes de software evolucionan, y en un cierto punto cada uno llega a un estado relativamente estable. Pero como los defectos se corrigen y las funciones de mejora se aplican, el resultado de esos cambios da como resultados nuevas versiones del software. Mantener el control de estas versiones de programas es a lo que llamamos control de versiones de componentes.

Un componente es identificado y etiquetado para diferenciarlo de los demás componentes en las versiones del software. Cuando un componente de software es modificado, las ediciones anteriores y las nuevas se pueden identificar por separado. Por lo tanto, cada versión, a excepción de la primera, tiene un predecesor. La sucesión de versiones de los software es la historia de sus diferentes componentes y su maleabilidad. Las diferentes versiones también funcionan como copias de seguridad de modo que uno puede regresar a las anteriores versiones del software.

Conclusiones

- ❶ Problemas que motivan la gestión de la configuración.
- ❷ Definiciones terminológicas.
- ❸ Qué es la trazabilidad y para que sirve.
- ❹ Implantación de planes para la gestión de la configuración.
- ❺ Identificación de versiones: árboles de características.
- ❻ Proceso de gestión de cambios: check-in, check-out.
- ❼ Concepto de auditoría.
- ❽ Concepto entregas y aspectos esenciales de la gestión de entregas.
- ❾ Problema de la construcción de entregas.

BIBLIOGRAFÍA.

- 1.- Roger Pressman, Ingeniería de Software un enfoque Práctico, Quinta y sexta Edición.

Complementaria.

- 2.- Ian Sommerville, Ingeniería del Software, Séptima Edición