

Gestión de la configuración del software



Material bibliográfico



- “Ingeniería del software. Un enfoque práctico”. Roger S. Pressman. 7ª edición. McGraw-Hill.
- “Software engineering”. Ian Sommerville. 10ª edición. Pearson.
- “Ingeniería del software. Aspectos de gestión. Tomo 1: Conceptos básicos, teoría, ejercicios y herramientas”. Román López-Cortijo y García y Antonio de Amescua Seco. Instituto Ibérico de la Industria del Software (www.iiis.es).
- “IEEE standard for software configuration management plans”. Estándar IEEE 828-1990.
- “IEEE guide to software configuration management”. Guía IEEE 1042-1987.
- “Interfaces, técnicas y prácticas. MÉTRICA versión 3”. Ministerio de las Administraciones Públicas: <http://www.csi.map.es/csi/metrica3/>.

Índice



- ¿Qué es la GCS?
- Objetivos de la GCS.
- Conceptos básicos de la GCS.
- Aspectos relacionados con la GCS.



¿Qué es la GCS?

¿Qué es la GCS?



- Según Babich:
 - El arte de coordinar el desarrollo de software para minimizar la confusión.
 - El arte de identificar, organizar y controlar las modificaciones que sufre el software que construye un equipo de programación.
 - El objetivo es maximizar la productividad minimizando los errores.
- Según IEEE:
 - La GCS cubre todas las actividades utilizadas para identificar y definir los elementos de configuración y sus relaciones.
 - Permite controlar cambios y modificaciones durante el ciclo de vida del software, conociendo los sucesivos estados del software que se archiva y la verificación de la completitud y consistencia de cada uno de estos estados.
- Según Pressman:
 - Analogía de la GCS con un restaurante en que la cocina tiene una puerta de entrada y otra de salida.
- Su misión se resume en:
 - Minimizar la confusión, minimizar los errores y maximizar la productividad.

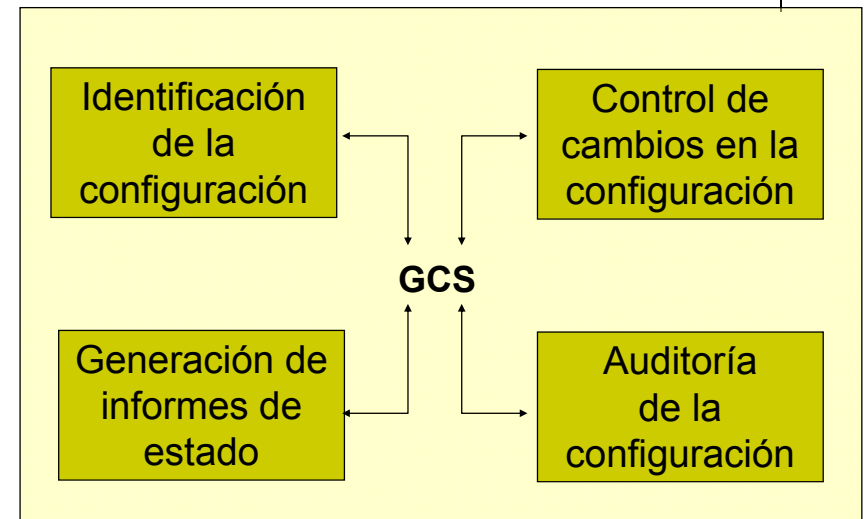
Objetivos de la GCS

Objetivos de la GCS



- Dos son los objetivos fundamentales de la GCS:
 - Facilitar la visibilidad:
 - Sobre el estado del producto: **estado**.
 - Sobre su historia: **evolución**.
 - Se evalúan y controlan los cambios.
 - Mantener la integridad del producto:
 - Establecer y mantener la integridad de los productos generados durante un proyecto y a lo largo de todo su ciclo de vida.
 - ¿Qué significa la integridad de un producto software? Significa que el producto cumple las siguientes condiciones:
 - Satisface las necesidades del usuario (requisitos del usuario, tanto los explícitos como los implícitos).
 - Cumple los requisitos de rendimiento.
 - Se puede trazar su evolución desde que se concibió y a través de todas las fases de su ciclo de vida.

¿Cómo conseguir los objetivos?





Conceptos básicos de la GCS

Definiciones básicas



- Configuración del software:
 - Conjunto de toda la información y productos utilizados o generados en un proyecto como resultado del proceso de ingeniería del software.
 - Por tanto, es el término que designa al conjunto de todos los elementos de configuración del software de un proyecto.
- Elemento de configuración del software (ECS):
 - Cada uno de los componentes de la configuración del software.
 - Es la unidad de trabajo para la GCS: Un ECS debe ser un elemento que se pueda definir y controlar de forma separada. Es decir, debe ser una unidad en sí mismo.
 - En cuanto al software propiamente dicho, dependiendo de su tamaño, complejidad y necesidad de control y visibilidad sobre el mismo, puede requerir de su descomposición en varios ECS, aunque el sistema en su conjunto es a su vez un ECS.

Línea base (baseline)



- Concepto introducido para facilitar el control de cambios:
 - Permitir cambios rápidos e informales sobre un ECS **antes** de que pase a formar parte de una línea base.
 - En el momento en que se establece una línea base, se debe aplicar un procedimiento formal para evaluar y verificar cada cambio.
 - Ejemplo: Un cambio en explotación en una librería de cálculo de la letra del NIF en un banco.
- Hay dos definiciones formales:
 - Desde el punto de vista del proceso:
 - Es un **punto de referencia** en el proceso de desarrollo que queda marcado por la **aprobación** de uno o más ECS mediante una **revisión técnica formal**.
 - Desde el punto de vista del producto:
 - Es un conjunto de **ECS revisados y aceptados** que sirven como base para el desarrollo posterior y que **sólo** se pueden cambiar a través de un **proceso formal de control de cambios**.

Aspectos relacionados con la GCS



Actividades relacionadas



- Las herramientas CASE que automatizan la GCS suelen incluir funciones adicionales que llevan a ampliar la definición estándar con las siguientes actividades:
 - Control de versiones:
 - Consiste en mantener un registro histórico de las diferentes versiones por las que pasan los componentes de un producto que permita la recuperación de cualquiera de ellas.
 - Construcción:
 - Consiste en gestionar la compilación y enlazado de los distintos componentes del producto software de una forma lo más eficiente posible.
 - Gestión de problemas:
 - Consiste en realizar un seguimiento de la evolución de los problemas que afectan al producto.
 - Control del trabajo en equipo:
 - Consiste en controlar las interacciones que se producen entre los múltiples desarrolladores de un producto, sobre todo cuando deben compartir ciertos componentes del producto.

Control de versiones



- Facilita la GCS.
- Permite saber para cada ECS:
 - Cuál es la última versión.
 - Relación entre distintas versiones (evolución de versiones).
 - Dónde están.
- Esto facilita el control de cambios:
 - ¿Sobre qué versión/es hacer un cambio?

Versiones y revisiones

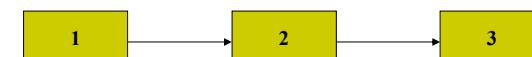


- Versión:
 - Es una instancia de un ECS, en un momento dado del proceso de desarrollo, que es almacenada en un repositorio y que puede ser recuperada en cualquier momento para su uso o modificación.
- Revisión:
 - A las distintas versiones que aparecen en el tiempo, según se va avanzando en el desarrollo de un ECS, se les suele llamar también revisiones.

Grafo de evolución de revisiones



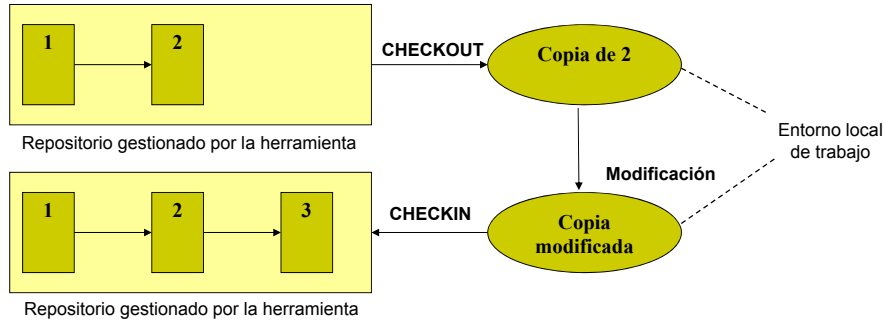
- Cada una de las revisiones de un ECS se debe poder identificar de manera única.
 - Es común utilizar para ello un esquema numérico, donde cada nueva versión recibe un número sucesivo.
- La manera más fácil de crear una nueva revisión de un ECS es realizar una modificación sobre la revisión más reciente. Así, las revisiones van formando una cadena, a la que se suele llamar cadena de revisión.
- Cada revisión en la cadena de revisión es una actualización de, y viene a sustituir a, la revisión anterior.
 - Normalmente se trabajará sobre la última revisión de la cadena, que es la más actual, aunque las anteriores también deben estar accesibles.
- Grafo de evolución o grafo de revisión:
 - Una representación para las diferentes revisiones de un ECS y sus relaciones de sucesión temporal:



Modelo de trabajo



- El modelo de trabajo de la mayoría de las herramientas de gestión de revisiones es :



- Las herramientas de gestión de revisiones o control de versiones ayudan a crear, identificar y almacenar nuevas versiones, al mismo tiempo que se mantienen las anteriores.

Almacenamiento de revisiones

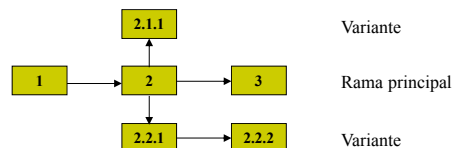


- Las herramientas de gestión de revisiones, por eficiencia, no almacenan físicamente todas las versiones.
 - Almacenan sólo una de ellas, que puede ser la primera o la última.
- Sin embargo, nos permiten recuperar cualquier otra versión.
 - Para ello guardan también toda la historia de cambios que han ocurrido sobre el elemento y que lo han hecho pasar de una versión a otra.
- Siendo r1 y r2 dos revisiones consecutivas en el grafo de evolución, se llama delta a:
 - La secuencia de operaciones que aplicadas sobre la revisión r1 dan como resultado la revisión r2.
- Tipos de deltas:
 - Según su dirección:
 - Deltas directos.
 - Deltas inversos.
 - Según su localización:
 - Deltas separados.
 - Deltas mezclados.

Variantes



- Son versiones de un ECS que coexisten en un determinado momento y que se diferencian en ciertas características.
- Representan la necesidad de que un objeto satisfaga distintos requisitos al mismo tiempo.
- Puede haber varias sobre las que se esté trabajando simultáneamente, a diferencia de las revisiones.
- Una variante no reemplaza a otra, como ocurre con las revisiones, sino que abre un nuevo camino de desarrollo.
 - Se reconocen fácilmente en el grafo de evolución como una ramificación de éste.



Tipos de variantes



- Las variantes pueden ser:
 - Temporales:
 - A veces es necesario que varias personas trabajen simultáneamente sobre la misma versión de un objeto y para que no ocurran conflictos entre ellas se crea una variante distinta para cada persona.
 - Una vez acabadas las modificaciones, es necesario mezclar todas las variantes para que la versión resultante contenga todos los cambios.
 - De usar y tirar:
 - Para explorar diferentes soluciones alternativas en paralelo y quedarse con la mejor.
 - Variantes de pruebas: Sobre las que se introducen elementos especiales para facilitar la realización de pruebas.
 - Permanentes, que se dividen en:
 - Variantes de requisitos de usuario: El caso más típico era el idioma en las aplicaciones.
 - Variantes de plataforma: Una variante por cada sistema operativo o plataforma hardware sobre la que se desee que funcione la aplicación.

Configuraciones alternativas



- Al abrir ramificaciones en el grafo de evolución, en vez de una única configuración vamos a tener un conjunto de configuraciones alternativas.
 - Cada una va a satisfacer las necesidades de un entorno particular o usuario.
- Cada configuración alternativa se especifica mediante los ECS que la componen y la versión adecuada de cada uno de ellos.
- Esto se puede conseguir de la siguiente forma:
 - Se asocian atributos a cada versión de un ECS.
 - Se crea una especificación de configuración que describa el conjunto de atributos deseado y se recuperan los ECS adecuados para construir la configuración.
- Make: Herramienta que facilita la construcción automática de una configuración concreta:
 - Recupera los ECS necesarios en la versión adecuada, los compila y los linka.

Release



- Se suele llamar release a una configuración del sistema que se va a comercializar o entregar al cliente.
- Debe identificarse y almacenarse para poder recuperarla en cualquier momento.
- La GCS también se encarga de controlar la gestión e instalación de releases.

Construcción (building)



- Esta actividad gestiona la compilación y el enlazado.
 - Es una actividad facilitada por la GCS.
 - Necesita saber:
 - Qué componentes enlazar.
 - En qué versión.
 - Dónde están.
 - Toma esta información de:
 - Identificación de la configuración.
 - Control de versiones.
 - Una vez que se han especificado las diferentes configuraciones del producto, existen herramientas que facilitan la construcción automática de una configuración concreta, es decir, la recuperación de los ECS necesarios en la versión adecuada, su compilación y enlazado.
 - Un ejemplo de este tipo de herramientas es make.

Gestión de problemas



- Es una actividad facilitada por la GCS y que se considera complementaria a la de control de cambios.
- El cambio sobre un producto puede venir dado por:
 - Cambio en los requisitos/necesidades.
 - Problema.
- Esta actividad gestiona la evolución de los problemas detectados sobre el software, tanto aquellos que se detectan en la fase de pruebas como los informes de problemas que llegan del usuario.
- Las tareas a realizar en la gestión de problemas son:
 - Admisión, registro y valoración de informes de incidencias.
 - Asignación del problema a un responsable.
 - Asociación de información al problema.
 - Monitorización del estado del problema.
 - Registro de actividades de corrección del problema.
 - Información acerca de los problemas (generación de informes, consultas a la base de datos de problemas y análisis estadísticos).

Control del trabajo en equipo



- Es una actividad facilitada por la GCS.
- Al compartir elementos de trabajo existe un claro y evidente peligro:
 - Sobreescritura de cambios.
- Solución:
 - Desarrollo en paralelo.
- Necesidad que se plantea ante esta solución:
 - Integración del trabajo realizado en paralelo:
 - Operación de integración (merge).

Otros aspectos relacionados



- La GCS tiene también una gran influencia en otros aspectos del desarrollo de software:
 - Las metodologías:
 - Integración de las actividades de GCS con las metodológicas.
 - Las fases que establezca la metodología, los productos que se generen, etc. son determinantes para establecer la GCS.
 - El entorno de desarrollo:
 - Uso de herramientas de GCS.
 - Integración con las otras herramientas del entorno de desarrollo.
 - La organización:
 - Aparecen nuevas políticas y procedimientos de GCS.
 - Aparecen nuevos roles y responsabilidades que deberán integrarse en la organización del proyecto.
 - La calidad:
 - Se contribuye a mantener la integridad del producto.