

# 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. 9ª edición. Addison-Wesley.
- “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](http://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



- Conceptos básicos en la GCS (SCM).
- Identificación de la configuración.
- Control de cambios en la configuración.
- Generación de informes de estado de la configuración.
- Auditoría de la configuración.
- Plan de gestión de la configuración.



# Conceptos básicos en la GCS

# ¿Qué es la GCS?



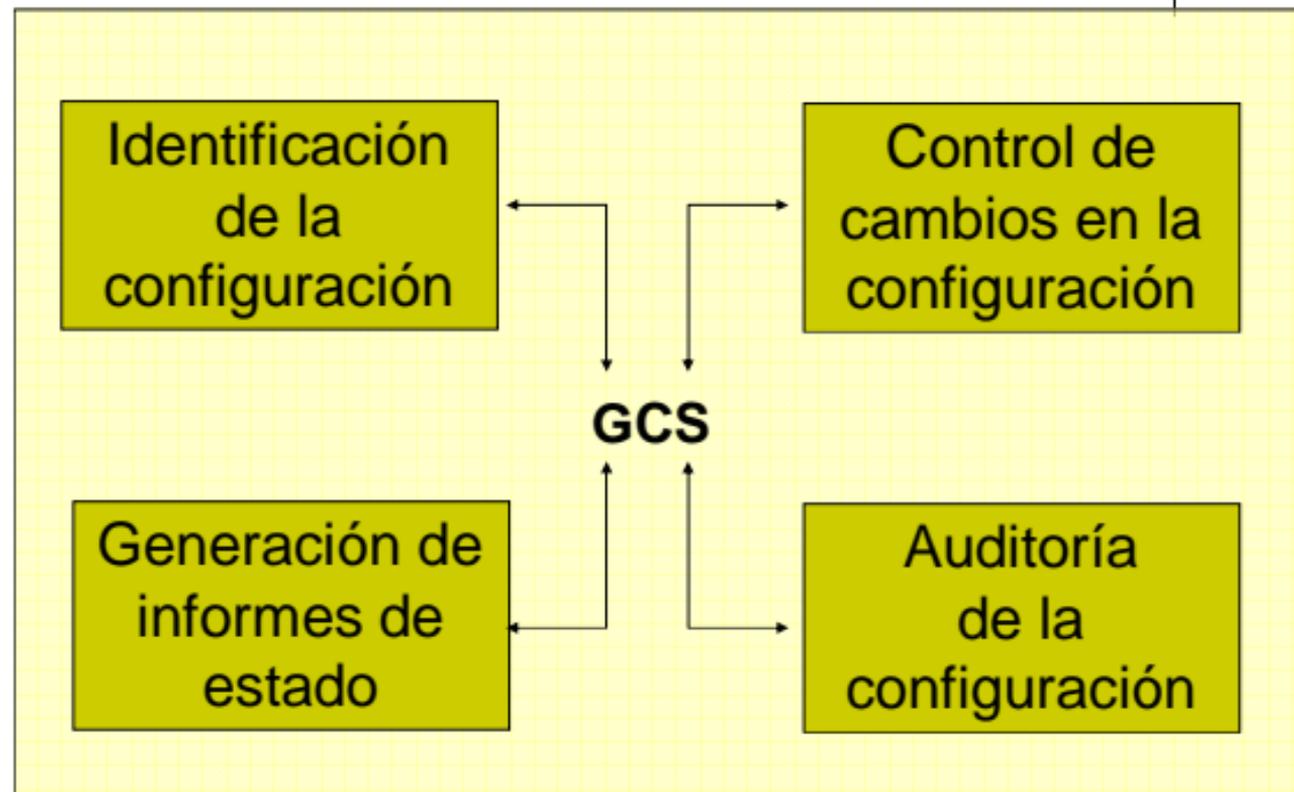
- Según Babich:
  - *El arte de coordinar el desarrollo de software para minimizar la confusión.*
  - *El arte de identificar, organizar y controlar las modificaciones que sufre el software que construye un equipo de programación.*
  - *El objetivo es maximizar la productividad minimizando los errores.*
- 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



- 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?



# 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**.

# 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



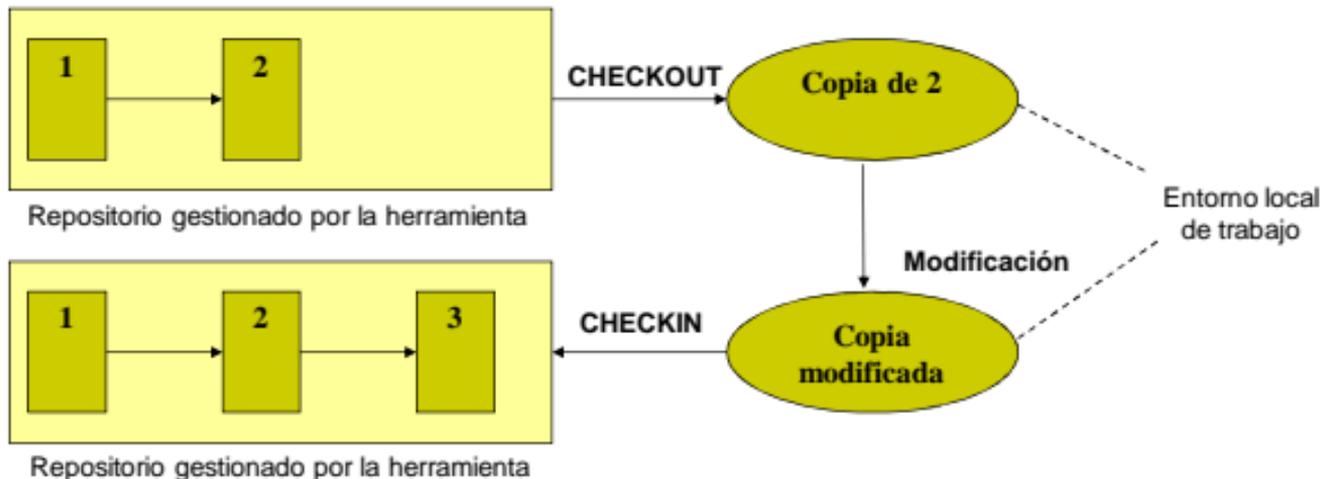
- Cada una de las revisiones de un ECS se debe poder identificar de manera única.
  - Es común utilizar para ello un esquema numérico, donde cada nueva versión recibe un número sucesivo.
- La manera más fácil de crear una nueva revisión de un ECS es realizar una modificación sobre la revisión más reciente. 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

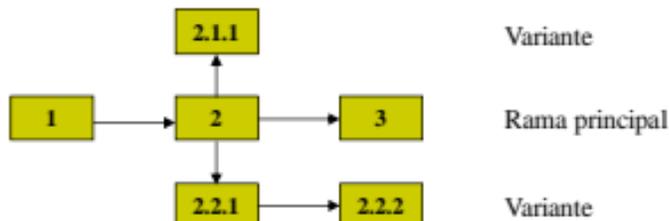


- Las herramientas de gestión de revisiones, por eficiencia, no almacenan físicamente todas las versiones.
  - Almacenan sólo una de ellas, que puede ser la primera o la última.
- Sin embargo, nos permiten recuperar cualquier otra versión.
  - Para ello guardan también toda la historia de cambios que han ocurrido sobre el elemento y que lo han hecho pasar de una versión a otra.
- Siendo  $r_1$  y  $r_2$  dos revisiones consecutivas en el grafo de evolución, se llama delta  $\alpha$ :
  - La secuencia de operaciones que aplicadas sobre la revisión  $r_1$  dan como resultado la revisión  $r_2$ .
- 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.



# Identificación de la configuración

# Objetivo



- La tarea de identificación de la configuración consiste en identificar y asignar nombres significativos y consistentes a todos y cada uno de los ECS.
- Esto conlleva determinar la visibilidad sobre el producto:
  - Identificar los ECS a considerar.
  - Identificar la estructura de dichos ECS.
- Es necesario proporcionar servicios de:
  - Identificación.
  - Accesibilidad.

# Actividades al inicio del proyecto



- La tarea de identificación engloba varias actividades:
  - Con respecto a los servicios de identificación:
    - Respecto a los ECS:
      - Identificación de la estructura y componentes del producto software.
      - Selección de los ECS.
      - Definición del esquema de identificación.
      - Selección de relaciones de configuración a mantener.
    - Respecto a las Líneas base (baselines):
      - Definición de líneas base.
  - Con respecto a los servicios de accesibilidad:
    - Definición de bibliotecas software.

# Actividades durante el proyecto

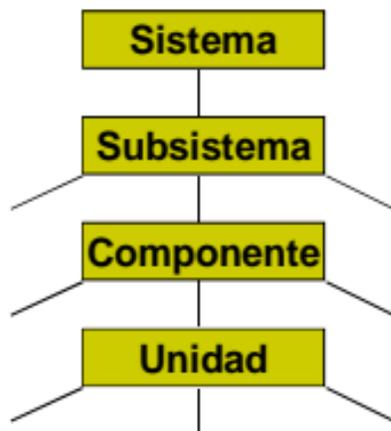


- Durante el desarrollo del proyecto se llevarán a cabo lógicamente las siguientes actividades:
  - Identificar/etiquetar los ECS generados.
  - Establecer las líneas base.
  - Mantener las relaciones de configuración.
  - Mantener las bibliotecas software.
- En la actividad de identificación de la configuración todavía no se tiene en cuenta el cambio.

# Identificación de estructura y componentes



- Se realizará, opcionalmente, al inicio del proyecto.
- Se trata de identificar la jerarquía del producto software:
  - Facilitará la ejecución de otras actividades posteriores de GCS (e.g., selección de los ECS y asignación de números de identificación).



- Es preliminar: Se irá concretando. Ahora se trata de obtener una primera visión de la estructura y los elementos que tendrá el sistema software.
- Lo importante es: ¿Con qué nivel de visibilidad se va a controlar?

# Selección de los ECS (I)



- Hay que considerar dos tipos de productos:
  - Productos generados por el desarrollo:
    - Indicados por la metodología seguida en el desarrollo.
    - Seleccionar los que van a estar bajo control de la GCS.
      - Los elegidos serán los ECS.
    - Decidir de qué forma se van a descomponer en otros.
  - Productos utilizados durante el desarrollo:
    - Hardware.
    - Software.
    - Manuales y documentación.
    - Estándares.
    - Normativas.

## Selección de los ECS (II)



- La selección de un número adecuado de ECS es muy importante.
- ¿Cuál es el número adecuado de ECS a identificar?
  - ¿Demasiados?
    - Inmanejable.
    - Costoso.
  - ¿Pocos?
    - Falta de visibilidad sobre el producto.

# Selección de los ECS: Ejemplo (I)



- Fase 0: Plan de Sistemas de Información:
  - Plan de Sistemas de Información.
- Fase 1: Análisis de Sistemas:
  - Módulo ARS: Análisis de Requisitos del Sistema.
    - Documento de Especificaciones del Sistema.
  - Módulo EFS: Especificación Funcional del Sistema.
    - Documento de Diseño Funcional.
      - Modelo de Procesos.
      - Modelo de Datos.
      - Modelo de Eventos.
    - Plan de Pruebas de Sistema.

# Selección de los ECS: Ejemplo (II)



- Fase 2: Diseño de Sistemas:
  - Módulo DTS: Diseño Técnico del Sistema.
    - Documento de Diseño Técnico.
      - Diseño de Datos: Esquema y contenido de BD.
      - Diseño Arquitectónico.
      - Diseño de los Módulos.
      - Diseño de los Interfaces de Usuario.
    - Documento de Operación.
      - Procedimientos de Operación.
      - Procedimientos de Seguridad y Control.
    - Plan de Pruebas.
      - Diseño de Pruebas de Integración.
      - Definición del Entorno de Pruebas.

# Selección de los ECS: Ejemplo (III)



- Fase 3: Construcción de Sistemas:
  - Módulo DCS: Desarrollo de Componentes del Sistema.
    - Código Fuente.
    - Programas Ejecutables.
    - Planes y Procedimientos de Prueba.
    - Casos de Prueba.
    - Informes de Prueba.
  - Módulo DPU: Desarrollo de Procedimientos de Usuario.
    - Manual de Usuario.
- Fase 4: Implantación de Sistemas.
  - Módulo PIA: Pruebas, Implantación y Aceptación del Sistema.
    - Plan de Pruebas, Implantación y Aceptación del Sistema.

# Esquema de identificación (I)



- Elementos que hay que identificar:
  - ECS.
  - Versiones.
  - Variantes.
  - Configuraciones alternativas.
  - Releases.
  
- Para identificar estos elementos es necesario un esquema de identificación ...

# Esquema de identificación (II)



- **Identificación unívoca de cada ECS:**
  - Definición del método que se va a utilizar para identificar de forma unívoca cada ECS: esquema de identificación para etiquetar los ECS.
  - Sea cual sea el esquema de identificación, éste debe proporcionar al menos la siguiente información:
    - Número o código del ECS.
    - Nombre del ECS.
    - Descripción del ECS.
    - Proyecto al que pertenece.
    - Fase/subfase de creación.
    - Fecha de creación.
    - Autor/es del ECS.
    - Tipo de ECS (documentos, programa, elemento físico, ...).
    - Localización.
    - Línea base a la que pertenece.
- **Identificación unívoca de cada versión de un mismo ECS:**
  - El esquema de identificación debe permitir diferenciar entre las diferentes versiones de un mismo ECS, para lo que debe proporcionar la siguiente información:
    - Número de versión.
    - Fecha de la versión.

# Esquema de identificación (III)



- Toda la información anterior puede ir codificada y agrupada en un único código o puede ser referenciada como partes separadas.
- A la hora de definir un código, se puede elegir entre dos tipos de sistemas de codificación:
  - No significativos (e.g., asignar números consecutivos):
    - Fácil asignación.
    - Difícil localización: Registro de asignación de códigos.
  - Significativos:
    - Fácil localización.
    - Interesante sólo si es fácil de recordar.
- ¿Dónde guardar los datos de identificación?:
  - Etiquetas en el propio ECS.
  - Fichas/listados.
  - Bases de datos.

## Esquema de identificación: Ejemplo (I)



- Se podría seguir un esquema de identificación como el siguiente, por poner un ejemplo:
  - Código ECS.
  - Nombre.
  - Descripción.
  - Proyecto.
  - Fase creación.
  
- Dada la siguiente forma de codificar y códigos ...
  - Código\_subfase + Código\_producto + Texto\_descriptivo.

## Esquema de identificación: Ejemplo (II)



- Código\_subfase:
  - PSI: Planificación de Sistemas de Información.
  - ARS: Análisis de Requisitos Software.
  - EFS: Especificación Funcional del Sistema.
  - DTS: Diseño Técnico del Sistema.
  - DCS: Desarrollo de Componentes del Sistema.
  - DPU: Desarrollo de Procedimientos de Usuario.
  - PIA: Pruebas, Implantación y Aceptación del Sistema.

## Esquema de identificación: Ejemplo (III)



- Código\_producto:
  - ERS\_RF: Requisito Funcional.
  - ERS\_RIU: Requisito de Interfaz de Usuario.
  - ERS\_RIH: Requisito de Interfaz Hardware.
  - ERS\_RIS: Requisito de Interfaz Software.
  - ERS\_RR: Requisito de Rendimiento.
  - ERS\_RD: Requisito de Desarrollo.
  - ERS\_RT: Requisito de Rendimiento.
  - ERS\_AS: Atributo de Seguridad.
  - ERS\_AM: Atributo de Mantenimiento.

## Esquema de identificación: Ejemplo (IV)



- Código\_producto:
  - MP\_DFD: Modelo de Procesos. DFD.
  - MP\_DD: Modelo de Procesos. Diccionario de Datos.
  - MP\_DP: Modelo de Procesos. Descripción de Proceso.
  - MD\_DER: Modelo de Datos. Diagrama E/R.
  - MD\_E: Modelo de Datos. Entidad.
  - MD\_R: Modelo de Datos. Relación.
  - ME\_CE: Modelo de Eventos. Catálogo de Eventos.
  - ME\_DFDE: Modelo de Eventos. DFD con eventos.
  - Etc.

# Esquema de identificación: Ejemplo (V)



- Un ejemplo de etiqueta de ECS podría ser:
  - Código ECS: ARS\_ERS\_RF\_G.Reservas.
  - Nombre: Requisitos de Gestión de Reservas.
  - Descripción: Requisitos específicos funcionales para la gestión de las reservas de entradas.
  - Proyecto: Teatro on-line.
  - Fase creación: Análisis de Sistemas.

- El contenido de este ECS podría ser:

## 3.1.9. Reserva de Localidades y Anulación de Reserva

### Introducción:

El sistema debe permitir la reserva de localidades y la anulación de reserva para cualquier representación. Una misma reserva puede incluir varias localidades, incluso en distintas zonas del teatro, pero para una misma representación.

### Entradas:

La entrada se corresponde con el documento Reserva descrito en el apartado 4.2.1.Documentos de Entrada.

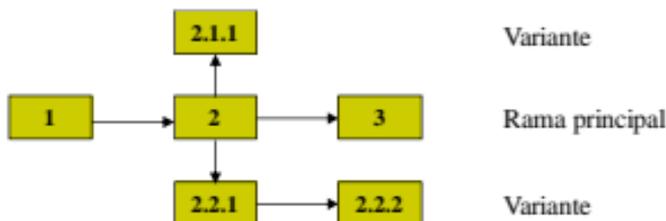
### Proceso:

Sólo se pueden crear reservas nuevas, no modificar las existentes...

# Esquema de identificación: Ejemplo (VI)



- Para considerar las versiones de un ECS:
  - Tabla de versiones:
    - Código del ECS.
    - Número versión.
    - Fecha creación.
    - Autores.
  - Identificación de versiones:
    - Nueva revisión: número versión + 1 en último campo
    - Nueva variante: número versión.(número última variante + 1). 1



## Esquema de identificación: Ejemplo (VII)



- Siguiendo con el ejemplo de Teatro on-line, el contenido de la tabla de versiones podría ser:
  - Código ECS: ARS\_ERS\_RF\_G.Reservas
  - Número versión: 1
  - Fecha creación: 14/07/2006
  - Autores: Pepe
  
  - Código ECS: ARS\_ERS\_RF\_G.Reservas
  - Número versión: 2
  - Fecha creación: 15/08/2006
  - Autores: Remedios
  
  - Etc.

# Relaciones en GCS



- Se puede considerar que los ECS son objetos y que están conectados con otros ECS mediante relaciones.
- Esta información ayudará al personal involucrado en la GCS a comprender:
  - Dónde se sitúa un ECS con respecto al resto.
  - El impacto de un cambio.
- Hay que tener en cuenta que el personal de GCS puede ser externo al equipo de desarrollo y necesita este tipo de ayudas para poder comprender los productos y el proceso de desarrollo.

# Relaciones interesantes en GCS



- Algunas relaciones especialmente interesantes a mantener son:
  - Composición (entre ECS):
    - Del software.
    - De la documentación.
  - Derivación (entre ECS):
    - Fuente – Objeto.
    - Caso de prueba – Traza de ejecución.
    - Diseño de módulo N – Fuente de módulo N.
  - Dependencia de cualquier tipo (entre ECS):
    - Modelo de datos – DFDs.
  - Sucesión (entre versiones de un ECS):
    - De versiones de un ECS.
  - Equivalencia (entre copias de una versión de un ECS):
    - Copia en disco = copia en papel = copia en cinta.

# Relaciones entre ECS (I)



- Sucesión:
  - Explícitamente: Tabla de sucesión:
    - Código ECS.
    - Versión antecesora.
    - Versión sucesora.
  - Implícitamente: En la numeración.
- Composición:
  - ECS contenedor.
  - ECS contenido.
- Derivación:
  - ECS origen.
  - ECS originado.
- Dependencia:
  - ECS uno.
  - ECS dos.

# Relaciones entre ECS (II)



- Equivalencia:
  - Tabla de copias:
    - Código ECS.
    - Versión.
    - Número copia.
    - Tipo: documento\_papel, manual, fichero, programa, máquina, cinta, ...
    - Localización: máquina + directorio, armario + caja, edificio + habitación, ...
  - Se añadirá el atributo “número de copias” en tabla de versiones.

# Definición de líneas base



- Las líneas base se establecen en hitos previamente especificados a lo largo del proceso de desarrollo.
  - Hay que definir cuáles van a ser esos hitos.
  - Lo normal es al final de determinadas fases para:
    - Identificar los resultados de las tareas de la fase.
    - Asegurar que se ha completado la fase.
- Decidir en cada proyecto:
  - Cuáles se van a establecer.
  - Cuándo.
  - Composición.
- Una línea base se puede establecer de dos formas:
  - Físicamente: Etiquetando cada ECS y almacenándolos en una biblioteca del proyecto.
  - Lógicamente: Publicando un documento de identificación de la configuración, que identifica el estado actual del producto en dicho punto del proceso de desarrollo.

# Líneas base más comunes



Funcional	Análisis de Requisitos del Sistema	Definición Problema Costes/ tiempos Requisitos Sistema
Distribución de Funciones	Análisis de Requisitos Software	ERS de cada componente software
Diseño Preliminar	Diseño arquitectura	Arquitectura Plan de Pruebas
Diseño	Diseño Detallado	Diseño detallado Plan de implementación Diseño de las pruebas
Producto	Pruebas	Programas Informes pruebas
Operación	Implantación	Manual usuario Manual instalación Manual operación

# Líneas base y composición: Ejemplo



- Con los mismos códigos de ejemplos anteriores, las líneas base podrían considerar los siguientes ECS:
  - Funcional: ERS.
  - Análisis Procesos: MP\_DFD, MP\_DD, MP\_DP.
  - Análisis Datos: MD\_DER, MD\_E, MD\_R.
  - Análisis Eventos: ME\_CE, ME\_DFDE.
  - Análisis: MP, MD, ME.
  - Diseño: DD\_T, DD\_CIBD, DA, DDM, DIE, DIU\_P, DIU\_I.
  - Producto: CF, CO, PE, FD.
  - Operación: MU, MO, MI.

# Bibliotecas de software



- Una biblioteca de software facilita la tarea de GCS, especialmente en cuanto al control de cambios y la generación de informes de estado.
- Es una colección controlada de software y/o documentación relacionada cuyo objetivo es ayudar en el desarrollo, uso o mantenimiento del software.
- Para cada proyecto hay que decidir:
  - Qué bibliotecas se van a usar.
  - Quién es el bibliotecario.
  - Procedimientos:
    - De introducción de elementos en la biblioteca.
    - De acceso a los elementos de la biblioteca.

# Bibliotecas de software: Ejemplo



- Área de Trabajo:
  - Elementos en desarrollo.
  - Cambio informal.
- Área de Integración:
  - Donde se juntan ECS para su integración en ECS de nivel superior.
  - No hay cambios.
- Biblioteca de Proyecto:
  - Una vez aprobada la línea base.
  - Cambio semiformal.
- Biblioteca Maestra:
  - Fin de proyecto y release.
  - Cambio formal.
- Repositorio SW:
  - Una vez retirado el producto.
  - No hay cambios.
- Repositorio de componentes reutilizables.



# Control de cambios en la configuración

# Control de cambios (I)



- Es la actividad de GCS más importante.
- Su objetivo es proporcionar un mecanismo riguroso para controlar los cambios.
- Normalmente combina procedimientos humanos y el uso de herramientas automáticas.
- Se consideran 2 tipos de cambios básicamente:
  - Corrección de defectos:
    - Los clientes tienden a clasificar todos los cambios en esta categoría.
  - Mejora del sistema:
    - Los programadores los suelen clasificar dentro de ésta.
  - Para solucionar el problema de determinar realmente de qué tipo es un cambio es fundamental la trazabilidad de los requisitos.

# Control de cambios (II)



- Se establecen varios niveles de control de cambios:
  - Informal:
    - Antes de que el ECS pase a formar parte de una línea base. Aquel que lo haya desarrollado podrá realizar cualquier cambio justificado sobre él.
  - Semiformal (o a nivel de proyecto):
    - Una vez que el ECS pasa la revisión técnica formal y se convierte en una línea base. Para que el encargado del desarrollo pueda realizar un cambio debe recibir la aprobación de:
      - El jefe de proyecto si es un cambio local.
      - El comité de control de cambios si tiene impacto sobre otros ECS.
  - Formal:
    - Se suele adoptar una vez que se empieza a comercializar el producto, cuando se transfieren los ECS a la biblioteca maestra. Todo cambio deberá ser aprobado por el comité de control de cambios.

# Comité de control de cambios



- Está formado por una o varias personas dependiendo del tamaño del proyecto y de la empresa.
- Tiene la autoridad sobre la aprobación, denegación y priorización de un cambio en un proceso formal o semiformal.
- Necesariamente debe tener una visión global del producto para poder evaluar el impacto (de cada cambio en un determinado ECS sobre otros ECS, sobre la calidad del producto, su rendimiento, su fiabilidad, sobre la visión que el cliente tiene, etc.).

# Responsabilidades en los cambios



- No sólo el comité es el responsable del control de cambios.
- También tienen responsabilidades los siguientes perfiles:
  - Todos los miembros de un proyecto.
  - El jefe de proyecto.
  - El bibliotecario.
- Es necesario establecer de forma precisa, al comienzo de cada proyecto, cuál será el proceso de gestión de cambios que se va a utilizar.

# Proceso formal de control (I)



1. Iniciación del cambio: Se presenta una solicitud de cambio, que puede venir provocada por un problema detectado o un cambio en los requisitos.
2. Clasificación y registro de la solicitud de cambio.
3. Aprobación o rechazo inicial de la solicitud de cambio. De ello suele ser responsable el comité de control de cambios.
4. Evaluación de la solicitud de cambio, si ha sido aprobada, para calcular el esfuerzo técnico, los posibles efectos secundarios, el impacto global sobre otras funciones del sistema y el coste estimado del cambio. Como resultado se obtiene un Informe de cambio.

# Proceso formal de control (II)



5. Se presenta el informe de cambio al comité de control de cambios. Si se considera beneficioso, se genera una orden de cambio (describe el cambio a realizar, las restricciones y los criterios de revisión y de auditoría) que se asigna a los desarrolladores. En este momento, el objeto a cambiar se da de baja en la biblioteca de soporte al proyecto.
6. Realización del cambio, seguimiento y control (actividad de gestión de problemas, si se debe a un problema).
7. Se certifica, mediante una revisión, que se ha efectuado correctamente el cambio y con ello se ha corregido el problema detectado o bien se han satisfecho los requisitos modificados. El objeto se devuelve a la biblioteca de soporte al proyecto.
8. Notificación al originador del cambio, que siempre se hará, aunque sea rechazado.

# Mecanismo de solicitud de cambios



- Formulario de solicitud de cambios:
  - Motivo del cambio.
    - Descripción del problema o cambio de requisitos.
  - Datos del solicitante.
  - Qué hay que cambiar.
  - Otros cambios relacionados o ECS afectados.
  - Alternativas.
- Informe de incidencia:
  - Fecha y hora de la incidencia.
  - Descripción.
  - Efectos.
  - Cómo replicar la incidencia.
  - Tipo de prueba que se realizaba.
  - Volcado de datos.
  - Datos del informante.



## **Generación de informes de estado de la configuración**

# Objetivo



- Mantener al tanto de:
  - El estado de la configuración.
  - Su evolución.
  
- A:
  - Desarrolladores.
  - Gestores.
  - Usuarios/clientes.
  
- En definitiva, pretende dar respuesta a las preguntas:
  - ¿Qué ocurrió?
  - ¿Cuándo ocurrió?

# Importancia



- Continuidad del proyecto:
  - Se trata de permitir que el proyecto siga adelante cuando, por ejemplo, el jefe de proyecto deja la empresa.
- Evitar duplicidad:
  - Si no se guarda información acerca de lo que ya se ha hecho, se puede estar repitiendo el trabajo ya hecho.
- Evitar repetir los errores repetidamente.
- Repetir lo que se hizo bien.
- Encontrar causas de problemas.
- Mejorar los problemas de comunicación entre los participantes en un proyecto.
  
- Esto se va a conseguir registrando toda la información necesaria acerca de lo que va ocurriendo y generando los informes necesarios.
  - Esta tarea implica, por tanto, tres actividades básicas:
    - Captura de la información.
    - Almacenamiento de la información.
    - Generación de informes.



# Actividades

- **Captura de la información:**
  - La información proviene de otras actividades de GCS.
  - La cantidad y tipo de información a capturar depende de las características del proyecto, su tamaño y complejidad.
  - El plan de GCS deberá indicar la información a capturar.
- **Almacenamiento de la información:**
  - Lo mejor suele ser almacenar esta información en una base de datos y utilizar herramientas automatizadas para gestionarla.
- **Generación de informes:**
  - Pertinentes.
  - De análisis post-mortem del proyecto.
  - Para estimaciones para proyectos futuros.

# Productos



- Los productos de esta actividad se agrupan fundamentalmente en dos categorías:
  - Registros.
  - Informes.
- Se deciden en cada proyecto particular a través del plan de gestión de la configuración.
  - Al comienzo de cada proyecto será necesario decidir qué tipo de registros se van a mantener y qué tipo de informes se van a generar y para quién.

# Ejemplos de registros (I)



- Identificación de la configuración:
  - De ECS.
  - De versiones y variantes.
  - De configuraciones alternativas.
  - De relaciones entre ECS:
    - Referencia a otros ECS.
  - De líneas base:
    - Referencia a los ECS que la componen.
  - De releases:
    - Fecha de liberación.
    - Composición: ECS + versión.
    - Diferencias release anterior.
  - De instalaciones:
    - Lugar.
    - Fecha.
    - Release.

# Ejemplos de registros (II)



- Control de cambios:
  - De incidencias:
    - Datos.
    - Resultado.
    - Historia.
  - De solicitudes de cambio:
    - Referencia a incidencias.
  - De cambios:
    - Referencia a solicitud de cambio.
    - Evaluación e impacto.
    - Disposición.
    - Plan de implementación.
    - Restricciones y criterios de revisión.
    - Historia del cambio: solicitud, aprobación/denegación.

# Ejemplos de registros (III)



- Control de cambios:
  - De modificaciones:
    - Referencia a solicitud de cambio o notificación de cambio.
    - Sobre el código.
    - Sobre la documentación.
    - Sobre bases de datos.
- Auditoría de la configuración:
  - Fecha.
  - Problemas detectados y recomendaciones.
- Actas de las reuniones del comité de control de cambios:
  - Asistentes.
  - Propósito.
  - Disposiciones.

# Informes



- Tipos según previsión:
  - Planificados.
  - Bajo demanda.
- Tipos según complejidad:
  - Directos: Contenido de los registros.
    - Inventario de ECS: Ofrece visibilidad sobre el contenido de las bibliotecas de proyecto.
    - Estado de los cambios: Es un resumen del estado en que se encuentran todas las solicitudes de cambio registradas durante un determinado período de tiempo.
    - Informe de incidencias: Es un resumen del estado en que se encuentran todas las incidencias originadas durante un determinado período de tiempo y las acciones a las que han dado lugar.
  - Indirectos: Suponen una mayor elaboración.
    - Diferencias entre versiones: Es un resumen de las diferencias entre las sucesivas versiones de un ECS.
    - Informe de modificaciones: Es un resumen de las modificaciones que se han efectuado en el producto software durante un determinado período de tiempo.



# Auditoría de la configuración

# Tipos de actividades de control de calidad



- Revisiones de fase:
  - Se realizan al final de cada fase y su objetivo es examinar los productos de dicha fase.
  - Las revisiones propias de la GCS son aquellas en que se establecerán las líneas base.
  - El objetivo de esta revisión es descubrir problemas, no ver que todo está bien.
- Revisiones de cambios:
  - Verificar que se han realizado correctamente los cambios aprobados sobre una línea base.
- Auditorías:
  - Se realizan al final del proceso de desarrollo de software y su objetivo es examinar el producto en su conjunto.

# Funciones en una revisión



- La tarea de revisión implica tres tipos de funciones:
  - Verificar la configuración actual del software con respecto a la línea base anterior.
    - Debe haber correspondencia y trazabilidad entre los ECS que aparecen en una línea base y los que aparecen en las líneas base que la preceden y que la siguen.
  - Validar que la configuración actual del software satisface la función que se esperaba del producto en cada hito del proceso de desarrollo.
    - Se realiza contra los requisitos.
  - Valorar si una línea base es aceptable o no, teniendo en cuenta los resultados de la verificación y validación, y otro tipo de comprobaciones.

# Auditorías



- Se suelen distinguir tres tipos de auditorías de configuración:
  - Funcional:
    - Objetivo: Asegurar que se han completado todas las pruebas necesarias para el ECS auditado y que, teniendo en cuenta los resultados de dichas pruebas, el ECS satisface los requisitos impuestos sobre él.
  - Física:
    - Inmediatamente después de la auditoría funcional.
    - Luego se establece la línea base de producto.
    - Objetivo: Verificar la adecuación, completitud y precisión de la documentación. Se trata de asegurar que representa el software que se ha codificado y probado.
  - Revisión formal de certificación:
    - Objetivo: Certificar que el ECS se comporta correctamente una vez que éste se encuentra en su entorno operativo.



# Plan de gestión de la configuración

# Estructura del plan según IEEE (I)



- **Introducción:**
  - Propósito del plan y a quién va dirigido.
  - Alcance: proyectos a los que se aplica, ECS bajo control, supuestos que podrían tener un impacto sobre la GCS ...
  - Definiciones y acrónimos, normalmente sólo del proyecto tratado.
  - Referencias a estándares IEEE, otros planes del proyecto, etc.
  - Definición de alto nivel del proceso de GCS.
- **Especificaciones de gestión para actividades de GCS:**
  - Organización: Contexto organizativo, relaciones de dependencia y autoridad ...
  - Responsabilidades: Comité de control, bibliotecario ...
  - Implantación del plan: establecimiento del comité de control, líneas base, revisiones y auditorías ...

# Estructura del plan según IEEE (II)



- Políticas, directivas y procedimientos aplicables a la GCS:
  - Niveles del software en un árbol jerárquico.
  - Nombrado de programas y módulos.
  - Designación de versiones.
  - Identificación de productos software.
  - Identificación de documentación.
  - Identificación de medios y ficheros.
  - Proceso de liberación de documentos.
  - Proceso de liberación de productos software.
  - Procesamiento de informes de incidencias, solicitudes de cambio, órdenes de cambio ...
  - Estructura y forma de operación de los comités de control.

# Estructura del plan según IEEE (III)



- Actividades de GCS:
  - Identificación de la configuración:
    - Descripción del esquema de identificación para los ECS.
    - Enumeración de las líneas base y para cada una de ellas se describe:
      - Momento de establecimiento.
      - ECS a incluir.
    - Bibliotecas y repositorios a utilizar, describiendo:
      - Procedimientos de inserción, recuperación, protección, etc.
  - Control de la configuración:
    - Mecanismos de iniciación de cambios:
      - Formularios.
      - Procedimientos.
    - Mecanismos de evaluación de cambios:
      - Procedimientos.
      - Criterios.
    - Mecanismos para aprobación/rechazo de los cambios:
      - Procedimientos.
      - Autoridad.
    - Mecanismos para verificación de cambios aprobados.
    - Mecanismos para gestión de problemas.
    - Mecanismos para control de versiones.

# Estructura del plan según IEEE (IV)



- Informes de estado de la configuración:
  - Registros a mantener.
  - Informes a generar.
  - Procedimientos de captura, almacenamiento y procesamiento de la información.
- Auditoría de la configuración: Descripción de cada una de las auditorías y revisiones que se van a realizar, indicando:
  - Objetivos.
  - ECS a auditar o revisar.
  - Participantes.
  - Procedimiento de realización.
  - Procedimiento para registrar deficiencias.
  - Listas de comprobación y cuestionarios a utilizar.
  - Criterios de aprobación del ECS.

# Estructura del plan según IEEE (V)



- Control de suministradores y subcontratistas: forma en que se va a controlar que los subcontratistas y suministradores satisfacen los requisitos de GCS establecidos.
- Recogida y retención de registros:
  - Qué documentación de GCS retener.
  - Métodos y recursos para recopilación, salvaguarda y mantenimiento de esta documentación, incluyendo cualquier tipo de Back-up.
  - Periodo de retención.