



## Ministerio de Educación y Deportes

Subsecretaría de Servicios Tecnológicos y  
Productivos



## Desarrollo de Software



Ministerio de  
Educación y Deportes  
Presidencia de la Nación



Ministerio de Producción  
Presidencia de la Nación

## Disciplinas de Gestión - Guía para la Construcción del Software



Disciplinas de Gestión

- Planificación de Proyecto
- Monitoreo y Control de Proyectos

**Estimar:** Predecir el tiempo y el costo que llevará desarrollar un producto desoftware, basándose en el tamaño de lo que se deseacntruir



# Disciplinas de Gestión - Guía para la Construcción del Software



Disciplinas de Gestión

- Planificación de Proyecto
- Monitoreo y Control de Proyectos

**Planificación de Proyecto:** Un plan es a un proyecto, lo que una hoja de ruta es a un viaje.

Planificar es definir qué es lo que haremos, cuándo lo haremos, cómo vamos a hacerlo y quién lo hará



## Planificación de Proyectos de Software

- La planificación y la estimación son críticas para el éxito de un proyecto
- Un plan de proyecto documenta:
  - Qué es lo que hacemos?
  - Cuándo lo hacemos?
  - Cómo lo hacemos?
  - Quién lo va a hacer?
- Los planes guían las decisiones de inversión
- La planificación es una tarea difícil, o no se planifica nada o se planifica algo que no se puede seguir

*Cuando el mapa y el territorio no coinciden..  
Confía en el territorio*





# Planificación de Proyectos de Software

- Una buena planificación ayuda a:
  - Reducir riesgos
  - Reducir incertidumbre
  - Soportar la toma de decisiones
  - Generar confianza
  - Transmitir información
- Los planes ayudan a saber quiénes están disponibles para trabajar
- Los planes ayudan a saber si un proyecto podrá entregar la funcionalidad que el usuario necesita y espera



## Definición de Alcance

### • **Alcance del Producto:**

- Son todas las características que pueden incluirse en un producto o servicio.

### • **Alcance del Proyecto:**

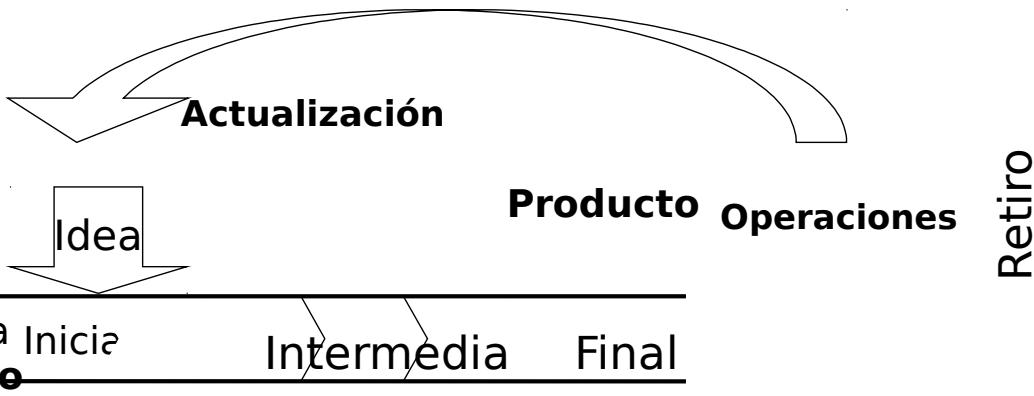
- Es todo el trabajo y solo el trabajo que debe hacerse para entregar el producto o servicio con todas las características y funciones especificadas.



# Relación: Ciclo de Vida del Proyectos y del Producto

Ciclo de Vida  
del **Producto**

Plan de  
Negocio



## Objetivos de la Planificación

Reducir  
Riesgos

Reducir  
incertidumbre

Soportar la  
toma de  
decisiones

Generar  
confianza

Transmitir  
información



## Planificación en el Contexto de Procesos de Control

Implica

Definir objetivos para el proyecto, que deben ser claros y alcanzables

Definir alcance para el proyecto, que está relacionado con el alcance del producto o servicio

Decidir el proceso y el ciclo de vida que se utilizará.

Definir roles y responsabilidades para cada uno de los miembros del equipo.

Estimar los recursos que serán requeridos.

Estimar tamaño, esfuerzo, tiempo y costo.

Identificar y realizar un análisis de riesgos.

Realizar el cronograma para el proyecto.

Definir el presupuesto.

Determinar el mecanismo de monitoreo y control y las métricas que se utilizarán.

Crear planes para las actividades de soporte, tales como administración de configuración y aseguramiento de calidad.



## Fallas de la Planificación

Fallas

Planificación por actividades no por características

Ley de Parkinson

La tardanza se trasmite en la programación

Actividades no independientes

Multitarea causa demoras adicionales

Las características no se entregan por prioridad

Ignoramos la incertidumbre

Estimaciones se vuelven compromisos



## Estimación de Software

“PREDICTION IS VERY DIFFICULT, ESPECIALLY ABOUT THE FUTURE.”

*La predicción es muy difícil, especialmente acerca del futuro.*

—NIELS BOHR,



## Estimación de Software

- La estimación es el proceso de encontrar una aproximación sobre una medida, lo que se ha de valorar con algún propósito
- En estadísticas estimación implica usar el valor de una estadística derivada de una muestra para determinar el valor de un parámetro correspondiente a una población
- Las estimaciones tienen asociado un valor de probabilidad, dado que no se realizan estimaciones en universos de certeza
- Se denomina “sobreestimar” si la estimación superó el resultado real y “subestimar” si la estimación fue de un valor inferior respecto del real
- Estimar software es predecir cuánto costará, cuánto durará un proyecto; la estimación tiene en cuenta objetivos de negocio, compromisos y control
- Las estimaciones y los planes están relacionados, sin embargo, estimar no es planificar y planificar no es estimar. La planificación es un proceso analítico e imparcial y el plan un proceso parcial en busca de objetivos



## Estimación de Software - Errores

Fuentes genéricas

Información imprecisa acerca del software a estimar

Información imprecisa acerca de la capacidad de la empresa que realizará el proyecto.

Demasiado caos en el proyecto.

Imprecisión generada por el proceso de estimación

Olvidar actividades puede provocar errores, no solo hay que tener en cuenta la programación sino tambien la gestión, hardware, administración, etc.



## Estimación en Procesos de Control Definidos



### Tamaño

- El Tamaño es lo primero que debe estimarse, cuán grande es? Qué vamos a construir?
- Es una de las estimaciones más difíciles de realizar
- Para estimar el tamaño se debe decidir qué contar:
  - En la captura de requerimientos: casos de uso, características, etc.
  - En análisis y diseño: puntos de función, pedidos de cambios, páginas web, reportes etc
  - Cuando el proyecto está más avanzado: defectos, clases, etc.



## Estimación en Procesos de Control Definidos



### Esfuerzo

- La estimación del Esfuerzo es la medida que determina el trabajo que una persona necesita realizar para obtener el producto de software , cuánto trabajo requerirá?
- La estimación de esfuerzo se deriva de la estimación de tamaño
- Se mide en horas hombre, cuánto tiempo le llevará a una persona realizar esta tarea?
- La experiencia de la persona que realizará esa tarea determinará el esfuerzo



## Estimación en Procesos de Control Definidos



### Tiempo

- Se debe estimar el tiempo calendario, cuándo estará terminado?
- Puede realizarse en horas, días, meses, años dependiendo del tamaño del producto a desarrollar
- Para estimar el tiempo hay que tener en cuenta:
  - Cuántas personas formarán el equipo de desarrollo?
  - Qué dedicación horaria diaria tendrá cada uno de los desarrolladores?
  - Cuántos días a la semana se trabajará?
  - Se pueden solapar actividades?
  - Cuán dependientes son las actividades a realizar por cada uno de los desarrolladores?



## Estimación en Procesos de Control Definidos



### Recursos

- No solo hay que tener en cuenta el esfuerzo de las personas sino también otros recursos como computadoras, software de base, herramientas, licencias etc.
- Hay recursos que si no están disponibles en tiempo y forma resultan críticos y puede representar un riesgo para el desarrollo del proyecto. Por ejemplo, un producto que se desarrollará para anteojos de google y no se disponen de los mismos para las pruebas



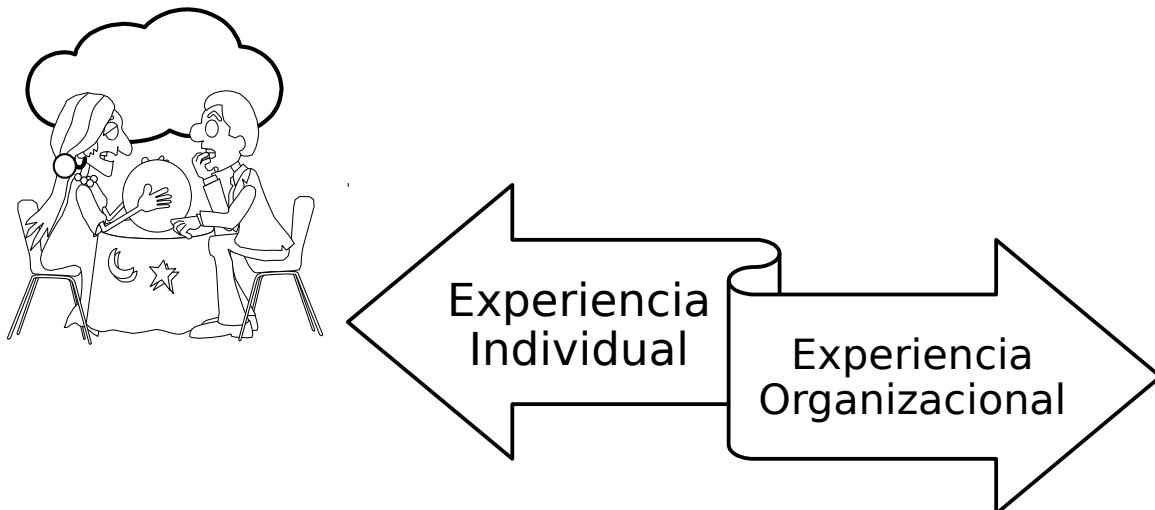
## Estimación en Procesos de Control Definidos



### Costos

- La estimación de costos asigna valor monetario al producto, cuánto costará?
- El costo más significativo es el del esfuerzo, el resto de los costos si bien se consideran son menos significativos
- Para determinar el costo del esfuerzo se debe considerar el costo de trabajo de cada uno de los desarrolladores y multiplicarlo por la cantidad de horas que se estimó de desarrollo
- Se debe tener en cuenta el costo de adquisición de recursos, viajes y viáticos, capacitación etc.

## Métodos de Estimación



## Métodos de Estimación

**Métodos heurísticos se basan en la experiencia, los principales son:**

- Basado en juicio experto individual: Comunmente llamado “a ojo”. Es el método más utilizado en la práctica, Lo usan alrededor del 75% de las organizaciones
  - Se pregunta a un experto su opinión
  - Se basa en los conocimientos del experto
  - Si desaparece el experto, la empresa deja de estimar
- Basado en un juicio experto grupal: El objetivo es buscar consenso basado en la discusión entre expertos. El método Delphi se basa en:
  - El anonimato de los participantes
  - La repetición y retroalimentación controlada
  - La respuesta del grupo en forma estadística
  - Las estimaciones en grupo suelen ser mejores que las individuales

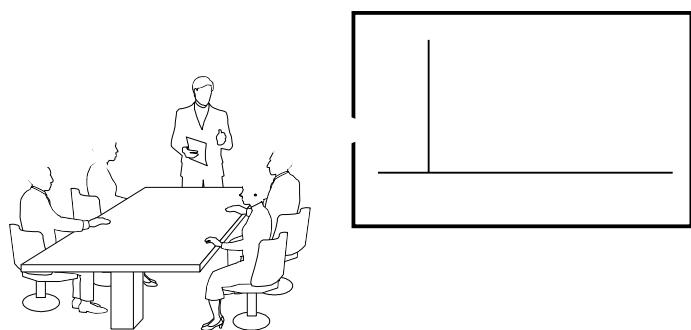


## Métodos de Estimación Wideband Delphi

- Basado en el método Delphi, requiere más interacción y discusión.
- Pasos
  - 1.Un coordinador presenta a cada experto una especificación y un formulario de estimación
  - 2.El coordinador convoca a una reunión de grupo en la que los expertos debaten



## Métodos de Estimación Wideband Delphi





## Métodos de Estimación Wideband Delphi

- Pasos

- 1.Un coordinador presenta a cada experto una especificación y un formulario de estimación
- 2.El coordinador convoca a una reunión de grupo en la que los expertos debaten
- 3.Los expertos llenan los formulario de forma anónima
- 4.El coordinador prepara y distribuye un resumen de las estimaciones



## Métodos de Estimación Wideband Delphi



Juan	*
Alicia	*
José	*
Maria	*
Estimaciones	



## Métodos de Estimación Wideband Delphi

### Pasos

1. Un coordinador presenta a cada experto una especificación y un formulario de estimación
2. El coordinador convoca a una reunión de grupo en la que los expertos debaten
3. Los expertos llenan los formulario de forma anónima
4. El coordinador prepara y distribuye un resumen de las estimaciones
5. El coordinador convoca a una reunión de grupo para concordar aquellos que no tenían las mismas estimaciones
6. Los expertos completan los formularios una vez más. Se repite 4 a 6 tantas veces como sea necesario



## Métodos de Estimación - Wideband Delphi



Juan	*
Alicia	*
José	*
Maria	*
Estimaciones	



Juan	*
Alicia	*
José	*
Maria	*
Estimaciones	



## Métodos de Estimación

- Consiste en comparar las especificaciones de un proyecto, con las de otros proyectos
  - Tamaño: ¿mayor o menor?
  - Complejidad: ¿Más complejo de lo usual?
  - Usuarios: Si hay más usuarios habrán más complicaciones.
  - Otros factores:
    - Sistema Operativo, entornos (la primera vez más)
    - Hardware, ¿Es la primera vez que se va a utilizar?
    - Personal del proyecto, ¿nuevos en la organización?



## Métodos de Estimación

### Método por analogía

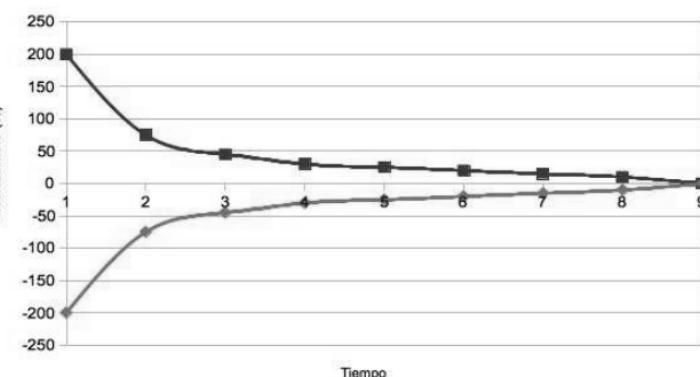
- Se basa en experiencias documentadas de cómo fueron los tiempos en proyectos previos
- Todas las experiencias se van almacenando para que sirvan de referencia a futuros proyectos

### Métodos paramétricos de estimación de software

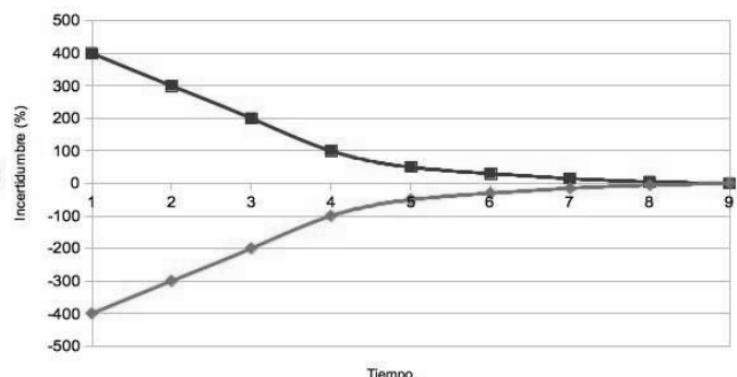
- COCOMO (Constructive Cost Model) II: Estima el esfuerzo basado en la funcionalidad, líneas de código etc.
  - SLIM (Software Lifecycle Management): Se usa un conjunto de fórmulas extraídas de grandes bases de datos de proyectos
- Estos métodos sirven para obtener una estimación del esfuerzo y basado en un cálculo previo del tamaño del software a desarrollar

## Incertidumbre en las Estimaciones

- En gestión el cono de la incertidumbre describe la evolución de la incertidumbre durante la ejecución de un proyecto
- Al principio se conoce poco sobre el producto y el resultado del proyecto, por lo que existe incertidumbre
- A medida que avanza el desarrollo se obtiene más información, por lo que la incertidumbre disminuye



Cono de incertidumbre en contexto estable



Cono de incertidumbre en desarrollo de Software

## Estimaciones en Contextos Inciertos

- Proveer estimación precisa en etapas tempranas tiene como consecuencia un compromiso poco probable de ser cumplido
- A medida que se adquiere conocimiento las estimaciones se harán más precisas
- Resulta difícil estimar cuando el ambiente es muy cambiante



# Disciplinas de Gestión - Guía para la Construcción del Software



Disciplinas de Gestión

- Planificación de Proyecto
- Monitoreo y Control de Proyectos

**Monitoreo y Control de Proyectos:** El monitoreo y control compara los planes realizados con el avance real de un proyecto. Da visibilidad respecto de la situación del proyecto en un momento de tiempo, cuánto hicimos, cuánto nos falta para terminar, cuánto gastamos, cuánto tiempo consumimos.



# Disciplinas de Gestión - Herramientas para la Calidad



Disciplinas de Soporte

- Gestión de Configuración de Software
- Aseguramiento de Calidad
- Métricas

**Gestión de configuración de software:** Una de las características del software es que es fácil de modificar, maleable y dado que vamos a necesitar cambiarlo muchas veces, es necesario crear mecanismos que nos ayuden a controlar el software a lo largo de su ciclo de vida .

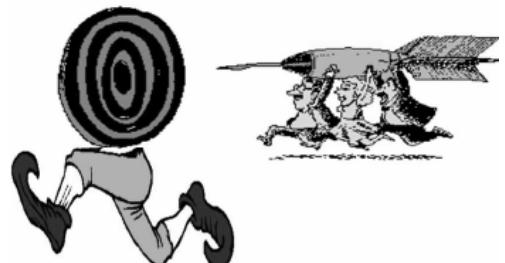




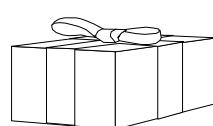
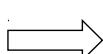
# Administrar la Configuración del Software

- La necesidad de gestionar la configuración surge del hecho que el software evoluciona constantemente
- Debido a que el software evoluciona es necesario mantener organizados y controlados los cambios que se realizan
- Los cambios en el software tienen su origen en:
  - Cambios del negocio 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

*El software: un blanco móvil*



## El Software



- Información:
  - estructurada con propiedades lógicas y funcionales.
  - creada y mantenida en varias formas y representaciones.
  - confeccionada para ser procesada por computadora en su estado más desarrollado

## Administrar la Configuración del Software

- Su propósito es establecer y mantener la integridad de los productos del proyecto de software a lo largo del ciclo de vida del mismo.
- Involucra para la configuración:
  - Identificarla en un momento dado
  - controlar sistemáticamente sus cambios
  - mantener su integridad y origen

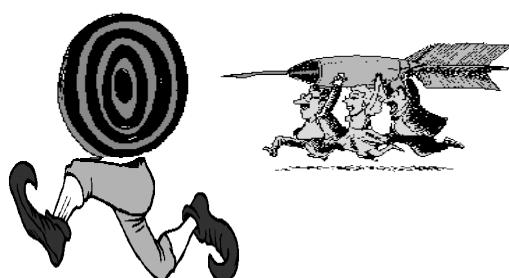


Una disciplina que aplica dirección y monitoreo administrativo y técnico a: identificar y documentar las características funcionales y técnicas de los ítems de configuración, controlar los cambios de esas características, registrar y reportar los cambios y su estado de implementación y verificar correspondencia con los requerimientos (ANSI/IEEE 828, 1990)

## Integridad del Producto

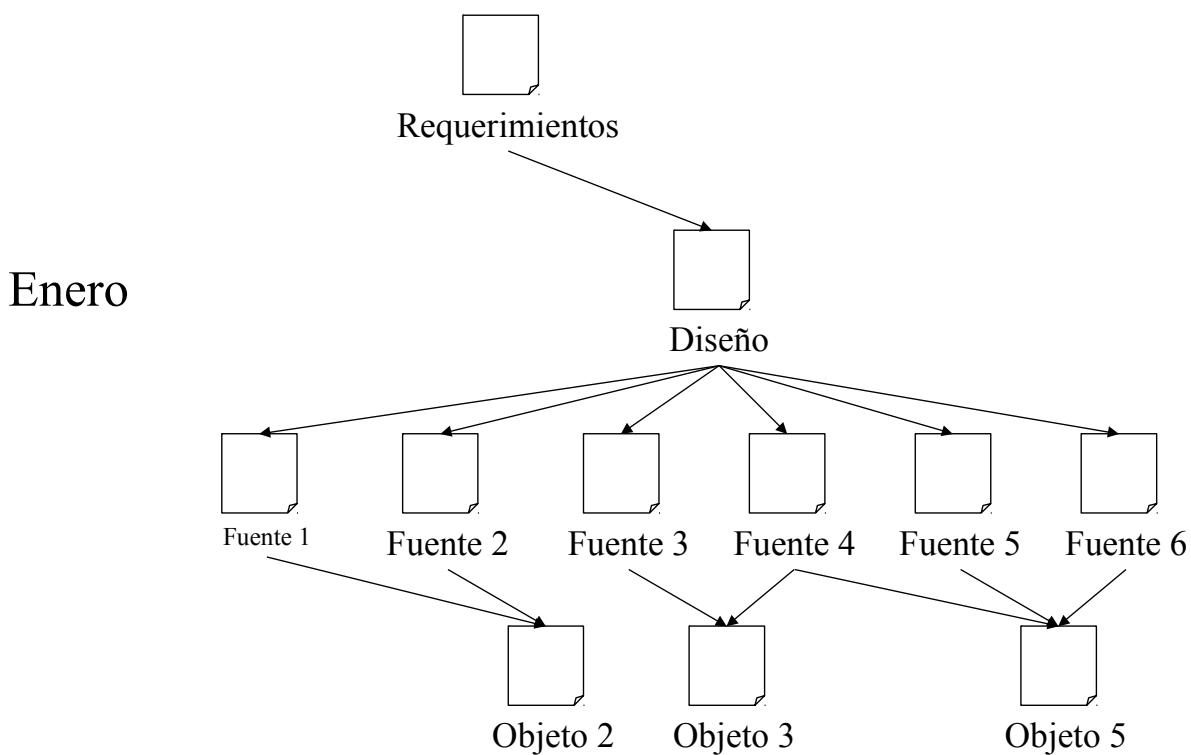
- Integridad
  - satisface las necesidades del usuario
  - puede ser fácil y completamente rastreado durante su ciclo de vida
  - satisface criterios de performance
  - cumple con sus expectativas de costo

El software: un blanco móvil

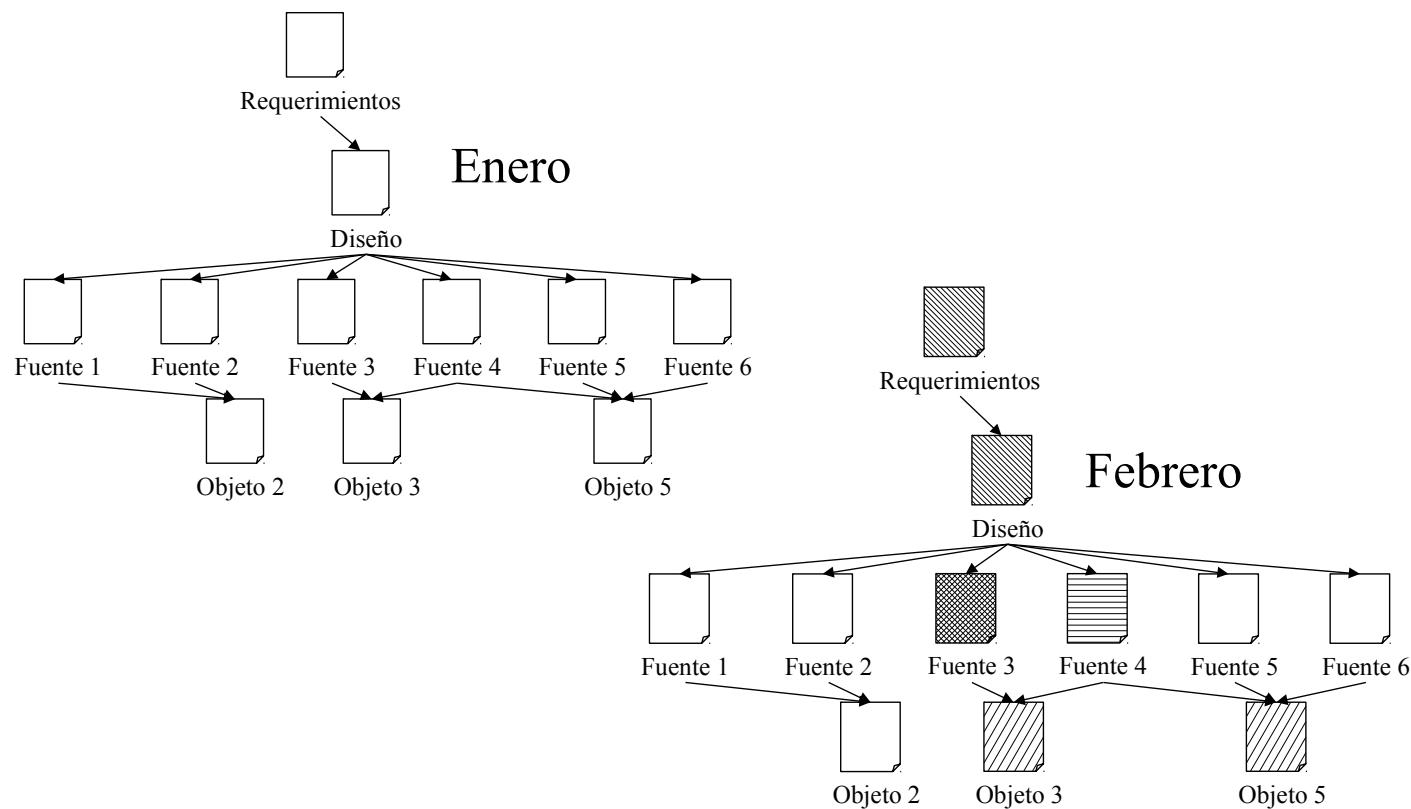




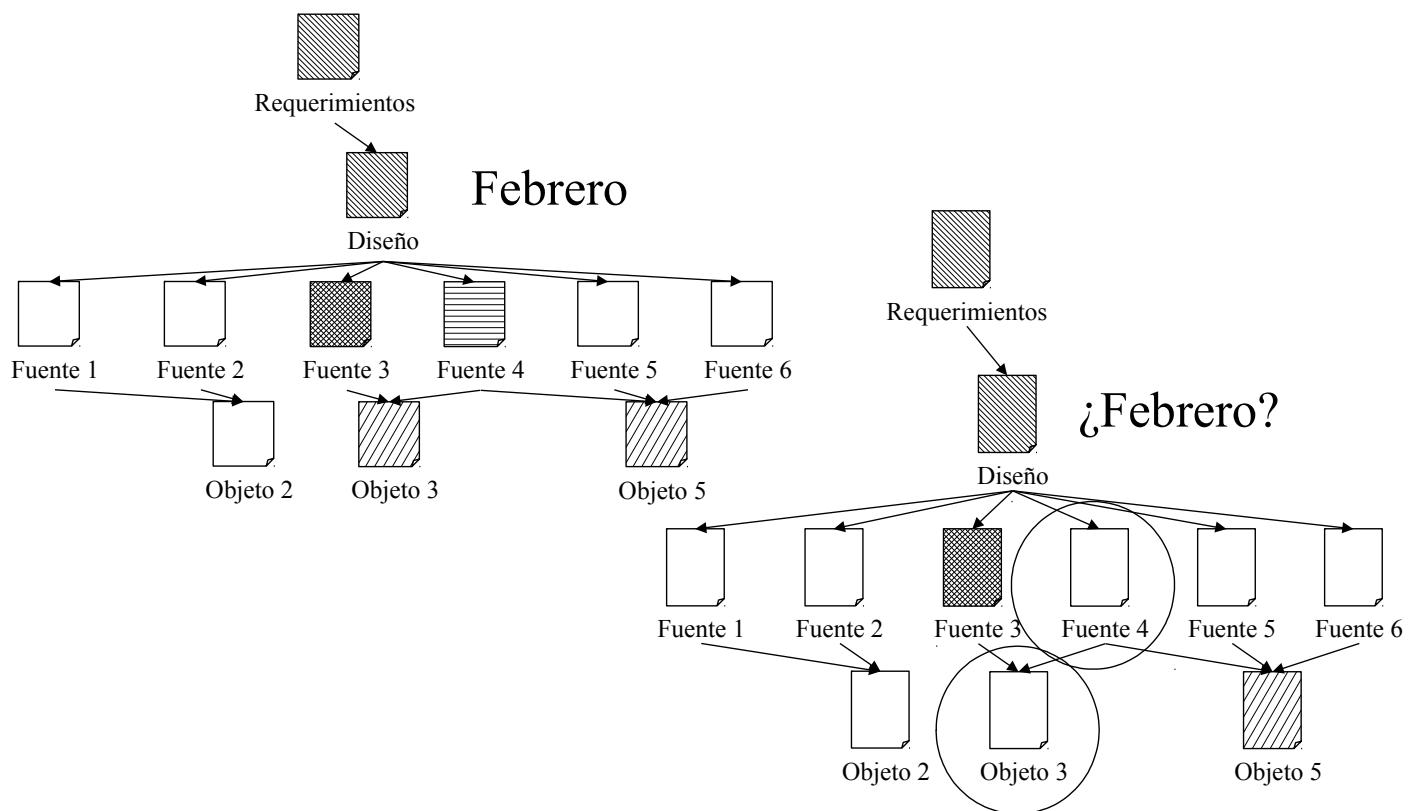
## El Software



## La Evolución del Software

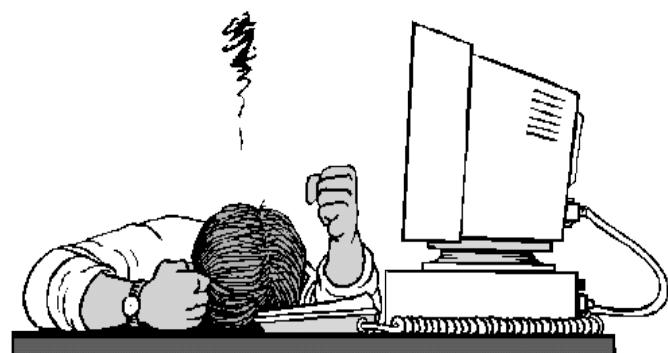


## Problemas en la Evolución del Software



## Problemas en el Manejo de Componentes

- Pérdida de un componente
- Pérdida de cambios (el componente que tengo no es el último)
- Doble mantenimiento
- Superposición de cambios
- Cambios no validados





# Conceptos de la Configuración del Software

## Configuración

- Conjunto de los componentes fuentes que son compilados en un ejecutable consistente, más todos los componentes, documentos e información de su estructura que definen una versión determinada del producto en un momento de tiempo
- Una configuración cambia porque se añaden, retiran o modifican elementos
- El control de configuración se refiere a la evolución de items de configuración
- La evolución de un sistema consiste en: añadir, suprimir, modificar items de configuración; o bien reorganizar la estructura
- Se llama:
  - Evolución temporal: *revisiones* Son cambios a lo largo del tiempo
  - Evolución espacial: *variantes* Son versiones (configuraciones) simultaneas



# Conceptos de la Configuración del Software

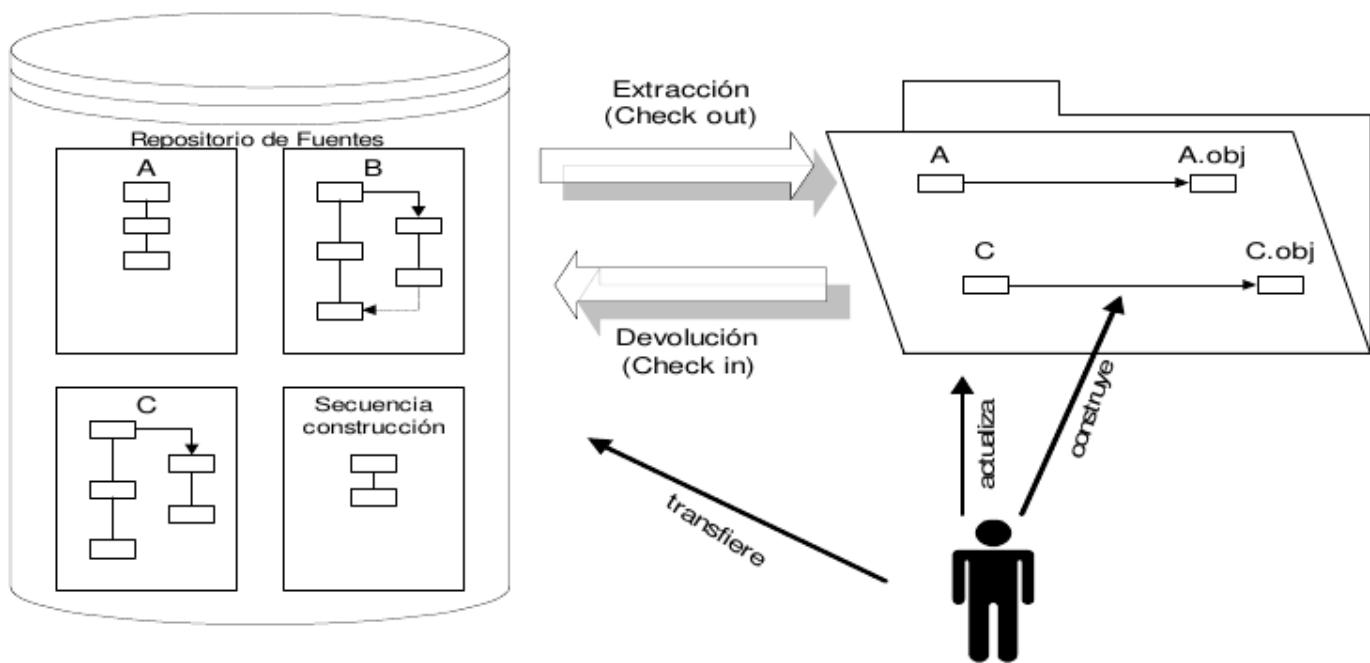
## Repositorio

- Un repositorio es un espacio de almacenamiento que contiene los items de configuración
- Puede estar conformado por una o varias bases de datos
- Permite centralizar el almacenamiento de los items de configuración (IC) de un mismo sistema o producto de software, incluyendo las distintas versiones de cada IC
- Mantiene la historia de cada CI con sus atributos y relaciones
- Usado para hacer evaluaciones de impacto de los cambios propuestos
- Permite ahorrar espacio de almacenamiento, evitando guardar por duplicado elementos comunes a varias versiones o configuración
- Facilita el almacenamiento de información de la evolución del sistema, no sólo de los IC en sí



# Conceptos de la Configuración del Software

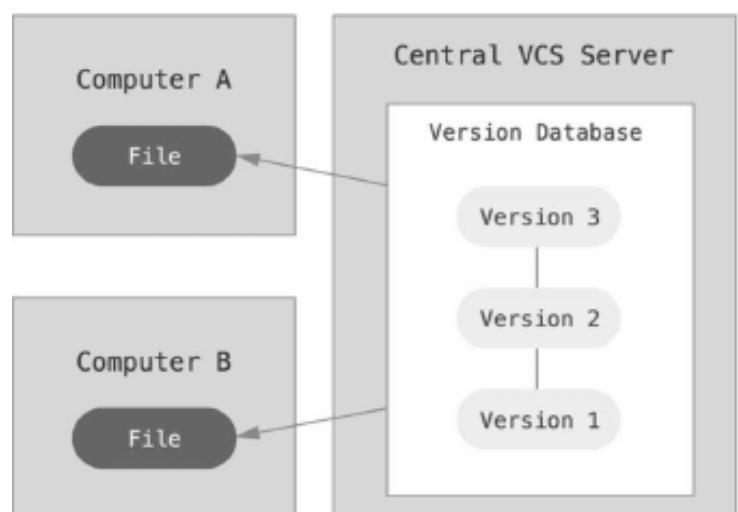
## Repositorio



# Conceptos de la Configuración del Software

## Repositorio Centralizado

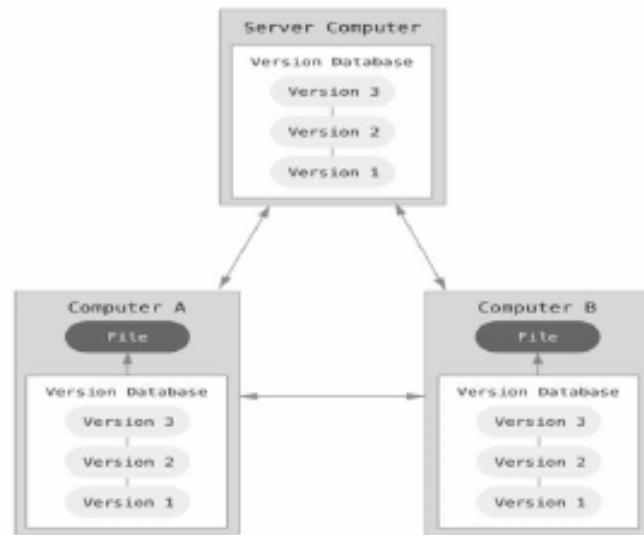
- Un servidor contiene todos los archivos con sus versiones
- Los administradores tienen mayor control sobre el repositorio
- Falla el servidor y nadie puede trabajar



# Conceptos de la Configuración del Software

## Repositorio Distribuido

- Cada cliente tiene una copia exactamente igual del repositorio completo
- Si un servidor falla sólo es cuestión de “copiar y pegar”
- Posibilita otras formas de trabajo no disponibles en el modelo centralizado



# Conceptos de la Configuración del Software

## Item de Configuración (IC)

- Un IC son todos y cada uno de los artefactos que forman parte del producto o del proyecto
- Pueden sufrir cambios o necesitan ser compartidos entre los miembros del equipo
- Ejemplos: documentos de requerimientos, de diseño, código fuente etc.



## Ejemplode de CI

- Plan de CM
- Propuestas de Cambio
- Visión
- 10 Riesgos principales
- Plan de desarrollo
- Prototipo de Interface
- Guía de Estilo de IHM
- Manual de Usuario
- Requerimientos
- Plan de Calidad
- Arquitectura del Software
- Plan de Integración
- Planes de fases
- Estándares de codificación
- Casos de prueba
- Código fuente
- Gráficos, iconos, ...
- Instructivo de ensamble
- Programa de instalación
- Documento de despliegue
- Lista de Control de entrega
- Formulario de aceptación
- Registro del proyecto



## Conceptos de la Configuración del Software

### Cambio

- Paso de una versión de la linea base a la siguiente

### Versión

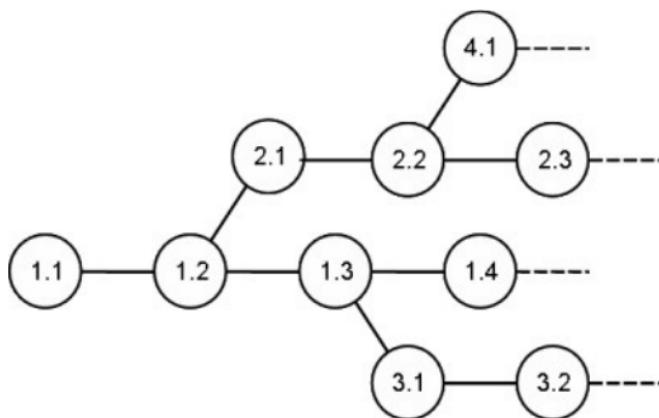
- Forma particular de un artefacto de un instante o contexto dado
- Pueden coexistir varsiones alternativas de un mismo artefacto
- El control de versiones se refiere a la evolución de un único IC o de cada IC por separado
- La evolución puede representarse gráficamente en forma de grafo, en el que los nodos son las versiones y los arcos la creación de una nueva versión



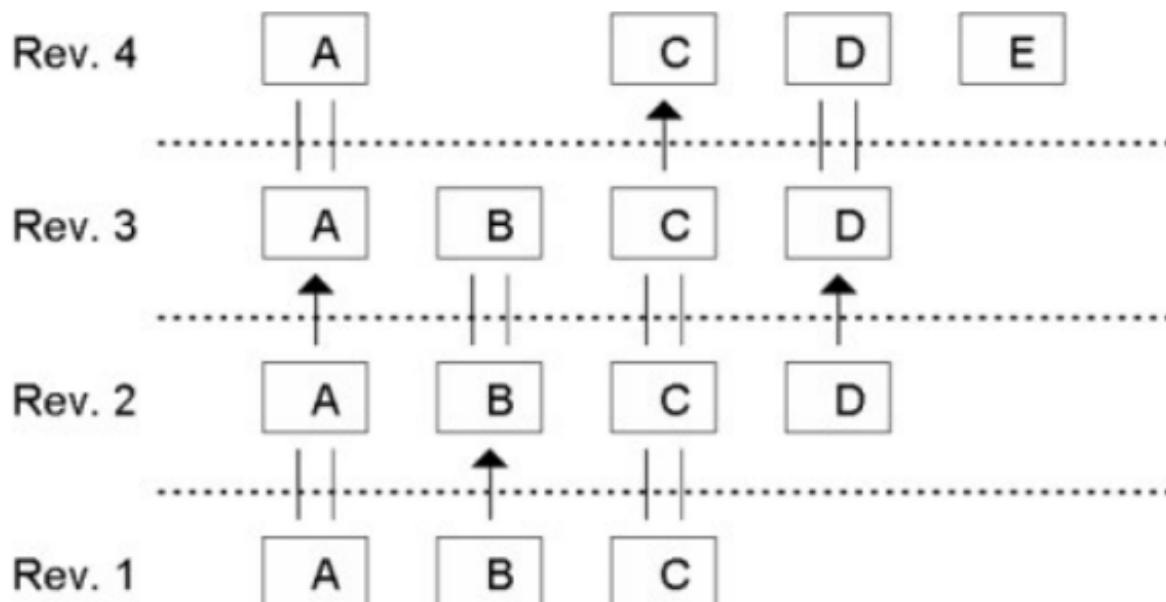
# Conceptos de la Configuración del Software

## Variante

- Una variante es una versión de un ítem de configuración que evoluciona por separado
- Representan configuraciones alternativas
- Un producto de software puede adoptar distintas formas (configuraciones) dependiendo del lugar donde se instale
- Cuando hay variantes, el grafo de evolución ya no es una secuencia lineal



# Ejemplo de Evolución de una Configuración





# Conceptos de la Configuración del Software

## Línea Base

- Una línea base es una configuración que ha sido revisada y sobre la que se ha llegado a un acuerdo
- Sirve como base para desarrollos posteriores y puede cambiarse a través de procedimientos formales de control
- El propósito de la creación y administración de líneas es el de poder retroceder en el tiempo y recuperar una versión anterior del producto

*Línea base es una especificación o producto que se ha revisado formalmente y sobre los que se ha llegado a un acuerdo, y que de ahí en adelante sirve como base para un desarrollo posterior y que puede cambiarse solamente a través de procedimientos formales de control de cambios Estándar IEEE 610.12-1990*

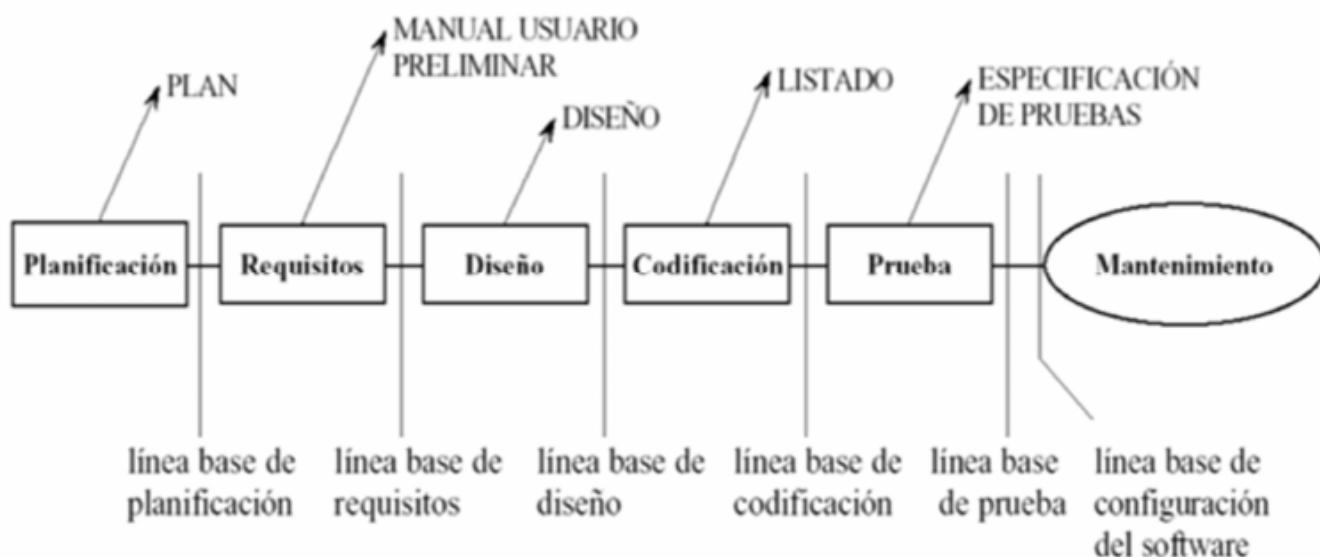


# Conceptos de la Configuración del Software

## Línea Base - Representación

Pueden ser:

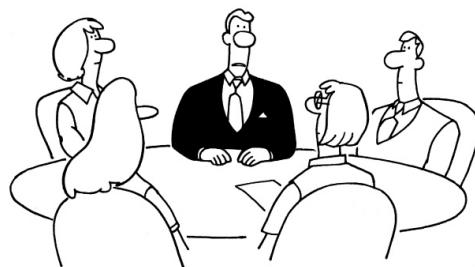
- De especificación (Requerimientos, Diseño)
- De productos que han pasado por un control de calidad definido previamente



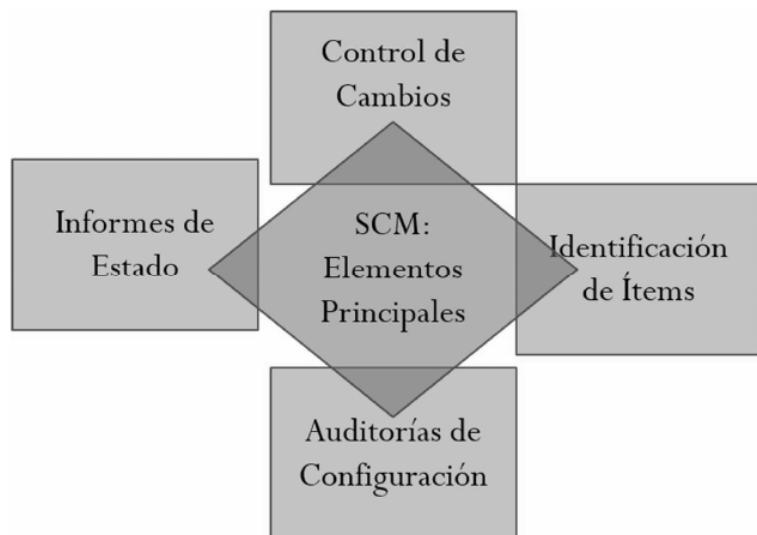
# Conceptos de la Configuración del Software

## Comité de control de cambios

- El comité de control de cambios es la autoridad de la aprobación de la incorporación de cambios en la linea base
- Se debe verificar que los cambios implementados son los que fueron autorizados
- Se debe tener un procedimiento para realizar esta tarea
- Está formado por representantes de todas las áreas del cambio propuesto
  - Líder de proyecto
  - Arquitecto
  - Analista funcional
  - Gestor de configuración de software
  - Desarrolladores
  - Analistas de pruebas
  - Representante del cliente



# Actividades en la Administración de Configuración del Software

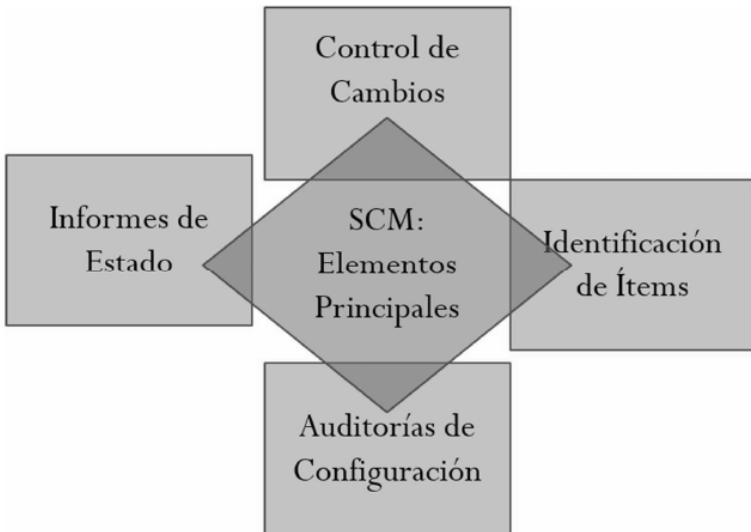


## Identificación de Ítems

- Se establecen estandares de documentación y un esquema de identificación para los ítems de configuración que se administrarán



# Actividades en la Administración de Configuración del Software

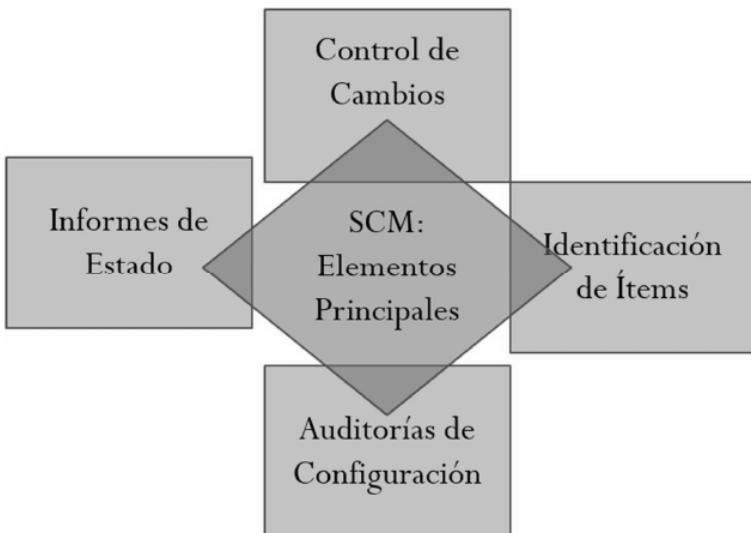


## Control de Cambios

- Consiste en la evaluación y registro de todos los cambios que se hagan a la configuración del software
- Surge debido a la continua evolución del software y la necesidad de registrar esos cambios en forma disciplinada



# Actividades en la Administración de Configuración del Software



## Auditoria de configuración

- Sirve para garantizar que el cambio se ha realizado correctamente
- Hay dos tipos de auditoría:
  - **Configuración física:** Asegura que lo indicado para cada IC en la linea base o en la actualización se ha alcanzado realmente
  - **Configuración funcional:** Evaluación de cada producto de software

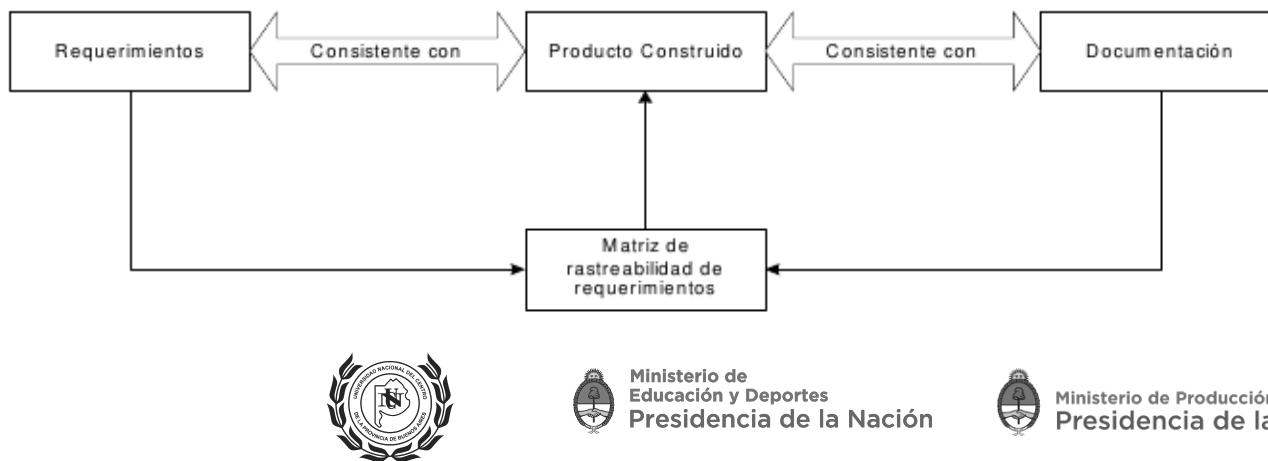
# Actividades en la Administración de Configuración del Software

## Auditoría de configuración

- Los dos tipos de auditorías se hacen a una línea base específica del producto en un momento de tiempo determinado
- Se controla la trazabilidad del producto, por lo que se usa una matriz de trazabilidad
- La trazabilidad plantea la incorporación de vínculos entre los ítems de configuración que permiten detectar si los requerimientos han

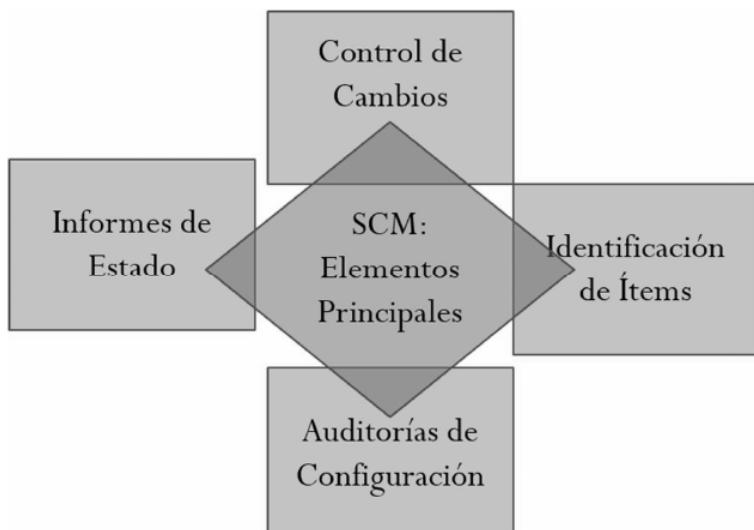
Auditoría Funcional de Configuración

Auditoría Física de Configuración



# Actividades en la Administración de Configuración del Software

## Informes de estado



- Se ocupa de mantener los registros de la evolución del sistema
- Algunos reportes que se pueden obtener son:
  - Momento de incorporación o actualización de una línea base
  - Estado de cada cambio propuesto
  - Momento en que se incorpora un cambio
  - Estado de la documentación administrativa o técnica
  - Información descriptiva de cada cambio propuesto

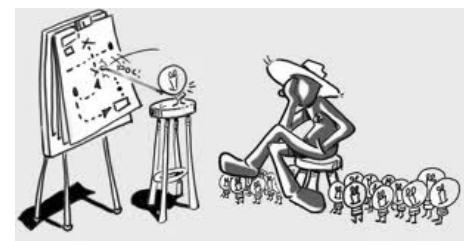
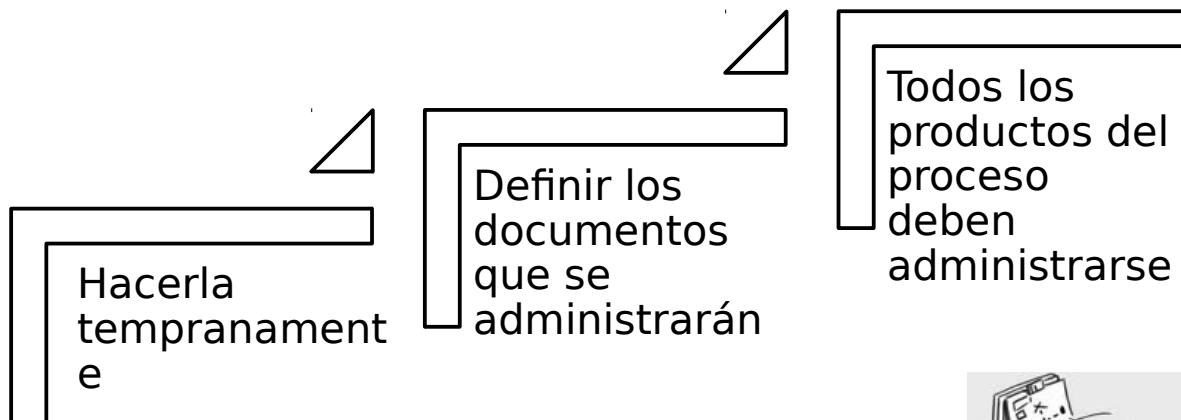


## Plan de Gestión de Configuración

- Las actividades relacionadas a la administración de configuración de software deben planificarse
- Las planificaciones deben hacerse lo más tempranamente posible
- La planificación debería incluir
  - Definición de los items de configuración (IC) que se administrarán
  - Reglas de nombrado de los IC
  - Herramientas a utilizar para la administración de configuración
  - Responsabilidades e integrantes del comité de control de cambios
  - Procedimiento formal de cambios
  - Procesos de auditoría
- Algunas herramientas existentes son:
  - MicroFocus -PVCS Dimensions
  - Platinum – CCC/Harvest
  - Rational – ClearCase
  - TortoiseSVN
  - Git



## Plan de Gestión de Configuración





# Tips para la Gestión de Configuración de Software

- Hacer de la administración de configuración de software (ACS) el trabajo de todos
- Crear un ambiente y un proceso de ingeniería que permita la administración de configuración
- Definir y documentar el proceso de administración de configuración/ingeniería, luego de seleccionas la/las herramientas que le den soporte al proceso
- El personal de ACS debe contar con individuos con conocimientos técnicos para dar soporte al desarrollo y mantenimiento del producto
- Los procedimientos y el plan de ACS deben realizarse en las etapas iniciales del proyecto



# Tips para la Gestión de Configuración de Software

- MicroFocus - PVCS Dimensions
- Platinum - CCC/Harvest
- Rational - ClearCase
- Microsoft - Team Fundation
- Accurev - Accurev
- Borland - StarTeam
- Herramientas de software libre para control de configuración
  - TortoiseSVN
  - Subversion
  - Git





## Disciplinas de Gestión - Herramientas para la Calidad



Disciplinas de Soporte

- Gestión de Configuración de Software
- Aseguramiento de Calidad
- Métricas

**Aseguramiento de Calidad:** El aseguramiento de la calidad del software es una actividad de protección que se aplica a lo largo de todo el proceso de construcción del software



## Disciplinas de Gestión - Herramientas para la Calidad



Disciplinas de Soporte

- Gestión de Configuración de Software
- Aseguramiento de Calidad
- Métricas

**Revisar técnicamente el software:** Tiene por objetivo detectar tempranamente errores que se cometen al desarrollar

Dado que el software es invisible, es esperable que posea errores





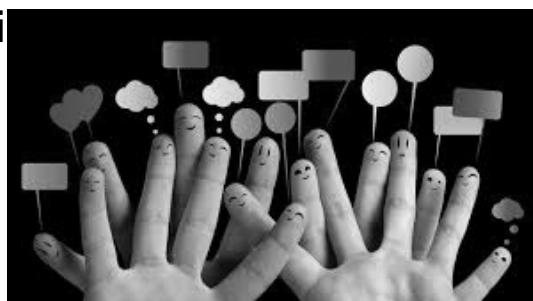
## Revisar Técnicamente el Software

- *El aseguramiento de la calidad del software es una actividad de protección que se aplica a lo largo de todo el proceso de construcción del software*
- Las revisiones técnicas surgen a partir de la necesidad de producir software de alta calidad
- Objetivos de las revisiones técnicas:
  - Descubrir errores en la función, lógica o implementación de cualquier representación del software
  - Verificar que el software bajo revisión alcanza sus requerimientos
  - Garantizar el uso de estándares predefinidos
  - Conseguir un desarrollo uniforme del software
  - Obtener proyectos que hagan sencillo los trabajos técnicos



## Revisar Técnicamente el Software

- Ventajas de las revisiones técnicas:
  - Reducción sustancial del costo del software, evitando re-trabajo
  - Gran valor educativo para los participantes
  - Sirve para comunicar la información técnica
  - Fomenta la seguridad y la conti



# Revisar Técnicamente el Software

- Directrices para la revisión:

- Revisar el producto y no al productor
- Hacer foco en los problemas no en la forma de presentación
- Indicar los errores en forma constructiva
- Fijar agenda y mantenerla
- Enunciar problemas no resueltos
- Limitar el debate y las impugnaciones
- Limitar el número de participantes
- Desarrollar una lista de comprobaciones para cada producto que pueda ser revisado
- Disponer de recursos y planificación de tiempos
- Entrenar los participantes
- Reparar las revisiones anteriores
- El problema debería ser resuelto por el autor



# Revisar Técnicamente el Software

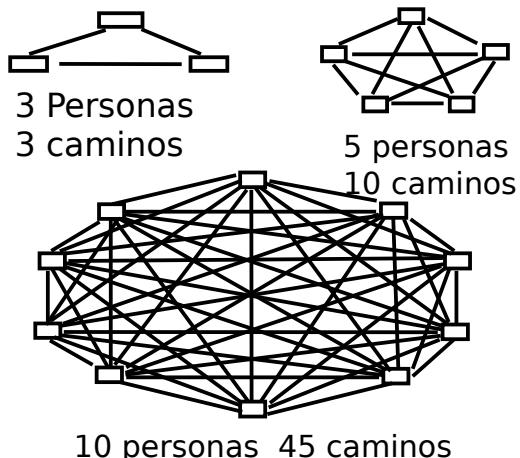
- Los artefactos generados durante el proyecto tienen revisiones asociadas

Artefacto de Software	Revisores sugeridos
Arquitectura o Diseño de alto nivel	Arquitecto, analista de requerimientos, diseñador, líder de proyecto, analistas de prueba.
Diseño detallado	Diseñador, arquitecto, programadores, analistas de prueba.
Planes de proyecto	Líder de proyecto, representante de ventas o marketing, líder técnico, representante del área de calidad.
Especificación de requerimientos	Analista de requerimientos, líder de proyecto, arquitecto, diseñador, analistas de prueba, representante de ventas y/o marketing.
Código fuente	Programador, diseñador, analistas de prueba, analista de requerimientos.
Plan de testing	Analistas de prueba, programador, arquitecto, diseñador, representante del área de calidad, analista de requerimientos.

## Defectos del Software - Fallas

- Los defectos o fallas son problemas de calidad descubiertos después de que el software haya sido entregado
- El objetivo primario de las revisiones técnicas es encontrar errores durante el proceso para evitar que se conviertan en defectos después de la entrega de software

### Ruido de comunicación

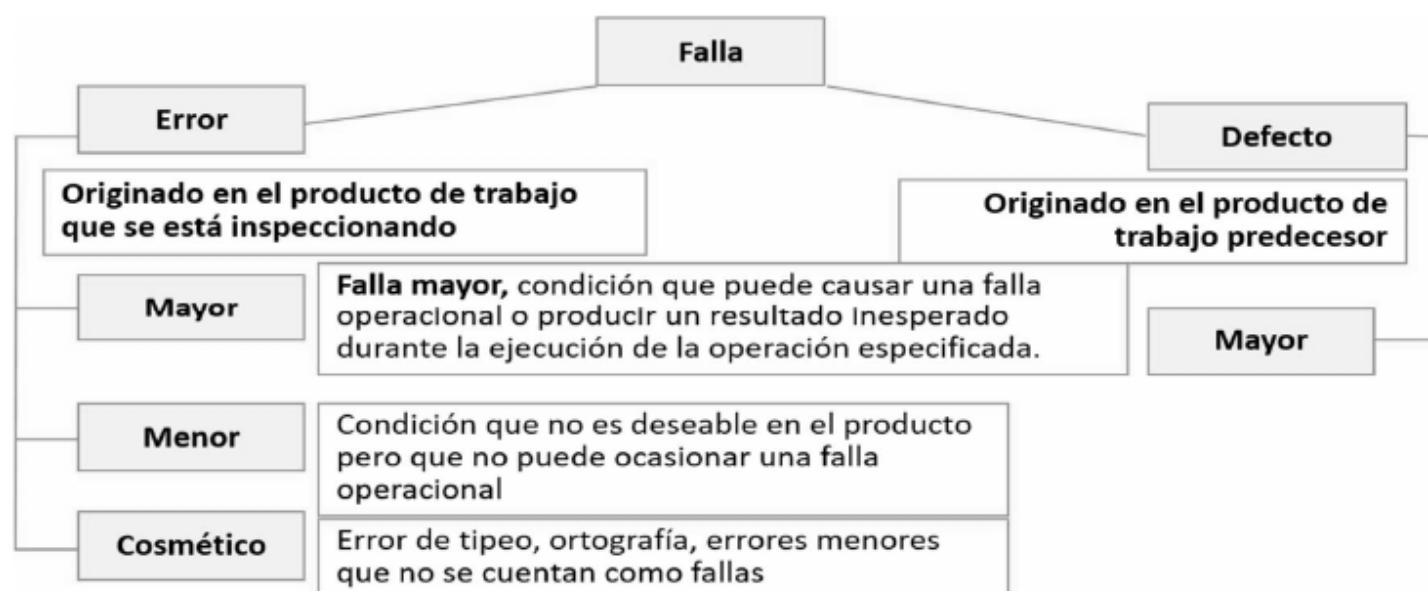


### Limitaciones de memoria

- Los límites de la memoria a corto plazo:  $7 +|- 2$
- “Las fallas más persistentes están relacionadas con la complejidad inherente al producto que se desarrolla”\*

\* Robert Glass, “Persistent Software Errors: 10 Years Later” 1st International Software Testing Analysis & Review Conference

## Defectos del Software - Fallas





## Revisión de Software

- Se ha demostrado que las revisiones de software son efectivas en un 75% a la hora de detectar errores
- Con la detección y eliminación de un gran porcentaje de errores se reduce el proceso de revisión
- Se puede resumir los beneficios de las revisiones técnicas en:
  - Reducción de los defectos en el uso del software
  - Reducción de los recursos de desarrollo, sobre todo en las etapas de codificación y prueba
  - Reducción en los costos de mantenimiento correctivo



## Revisiones Técnicas - Inspección de Software

*Es una actividad de aseguramiento de calidad del software*

Objetivos:

- Descubrir errores
  - Verificar que el software alcanza sus requisitos
  - Garantizar que el software ha sido representado de acuerdo a ciertos estándares
  - Conseguir un software desarrollado de manera uniforme
  - Hacer que los proyectos sean más manejables
- 
- Se lleva a cabo mediante una reunión y el éxito depende de su planificación





# Revisiones Técnicas - Inspección de Software

## SON

- La forma más barata y efectiva de encontrar fallas
- Una forma de proveer métricas al proyecto
- Una buena forma de proveer conocimiento cruzado
- Una buena forma de promover el trabajo en grupo
- Un método probado para mejorar la calidad del producto

## NO SON

- Utilizadas para encontrar soluciones a las fallas
- Usadas para obtener la aprobación de un producto de trabajo
- Usadas para evaluar el desempeño de las personas



## El Proceso de Inspección de Software

- Se puede ver al proceso de inspección de software como un repaso detallado y formal del trabajo en progreso. Pequeños grupos (4 o 5) estudian el “producto de trabajo” (líneas de código, requerimientos etc) independientemente y luego se reúnen.

Rol	Responsabilidad
Autor	<ul style="list-style-type: none"><li>• Creador o encargado de mantener el producto que va a ser inspeccionado.</li><li>• Inicia el proceso asignando un moderador y designa junto al moderador el resto de los roles.</li><li>• Entrega el producto a ser inspeccionado al moderador.</li><li>• Reporta el tiempo de trabajo y el nro. total de defectos al moderador.</li></ul>
Moderador	<ul style="list-style-type: none"><li>• Planifica y lidera la revisión.</li><li>• Trabaja junto al autor para seleccionar el resto de los roles.</li><li>• Entrega el producto a inspeccionar a los inspectores con tiempo (48hs) antes de la reunión.</li><li>• Coordina la reunión asegurándose que no hay conductas inapropiadas</li><li>• Hacer seguimiento de los defectos reportados.</li></ul>
Lector	Lee el producto a ser inspeccionado.
Anotador	Registra los hallazgos de la revisión
Inspector	Examina el producto antes de la reunión para encontrar defectos. Registra sus tiempos de preparación.



# El Proceso de Inspección de Software

## Planificación



## Visión General



## Preparación



## Reunión de Insp.



## Corrección



## Seguimiento

- **Planificación:** Elegir equipo, preparar material y calendario. Se completa el producto de trabajo y un moderador lo asigna al inspector que cumple el rol correspondiente
- **Visión Gral:** Presentar proceso y producto. Si los inspectores no están familiarizados con el software se puede hacer una reunión general
- **Preparación:** Análisis individual para encontrar potenciales defectos.
- **Reunión:** Los inspectores se reúnen para analizar su trabajo individual en forma grupal para recolectar potenciales defectos previos, filtrar falsos positivos
- **Corrección:** El autor corrige los defectos
- **Seguimiento:** El moderador chequea las correcciones del autor



## El Proceso de Inspección de Software - Puntos claves

- El proceso de inspección debe ser llevado a cabo por personas que conozcan tanto del dominio específico, del producto de software, así como de la tecnología aplicada a las soluciones que serán objeto de la inspección
  - Revisar el producto, no al productor
  - Fijar una agenda y cumplirla
  - Limitar el debate y las impugnaciones
  - Enunciar las áreas de problemas, pero no tratar de resolver cualquier problema que se manifieste
  - Tomar notas escritas
  - Limitar el nro. De participantes e insistir en la preparación por anticipado
  - Desarrollar una lista de revisión
  - Disponer recursos y una agenda
  - Entrenamiento
  - Repasar revisiones anteriores
  - Aprender sobre inspecciones y convencer al proyecto de utilizarlas
  - Determinar en qué parte del proyecto deben ser utilizadas
  - Definir para cada proyecto cómo se realizarán las revisiones técnicas



# Disciplinas de Gestión - Herramientas para la Calidad



Disciplinas de Soporte

## Métricas:

COMPLETAR!

- Gestión de Configuración de Software
- Aseguramiento de Calidad
- Métricas