



# Ingeniería de software II

Gestión de la Configuración del Software

Gestión de Proyectos



# Contenidos

---

- » Gestión de la Configuración del Software
- » Gestión de Proyectos
  - Planificación Temporal
  - Planificación Organizativa

2



# Gestión de la Configuración del Software (GCS)

---

## » ¿Qué es un software?

Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación

» Gestión de Configuración es el proceso de identificar y definir los elementos en el sistema, controlando el cambio de estos elementos a lo largo de su ciclo de vida, registrando y reportando el estado de los elementos y las solicitudes de cambio, y verificando que los elementos estén completos y que sean los correctos.

» Es una actividad de autoprotección que se aplica durante el proceso del software.

3



# Gestión de la Configuración del Software (GCS)

- »El resultado del proceso de Software se puede dividir en:
- Programas (códigos y ejecutables)
  - Documentos
  - Datos

Elementos de la configuración (ECS)

Avanza el Proyecto

ECS - Cambian constantemente

Control muy exhaustivo de esos cambios

GCS

4



# Gestión de la Configuración del Software

## Elementos de la GCS - (ECS)

---

1. Especificación del sistema
2. Plan del proyecto software
3. a) Especificación de requerimientos del software  
b) Prototipo ejecutable o en papel
4. Manual de usuario preliminar
5. Especificación de diseño:  
a) Diseño preliminar  
b) Diseño detallado
6. Listados del código fuente
7. a) Planificación y procedimiento de prueba  
b) Casos de prueba y resultados registrados
8. Manuales de operación y de instalación
9. Programas ejecutables  
a) Módulos, código ejecutable  
b) Módulos enlazados
10. Descripción de la base de datos  
a) Esquema, modelos  
b) Datos iniciales
11. Manual de usuario
12. Documentos de mantenimiento  
a) Informes de problemas del software  
b) Peticiones de mantenimiento  
c) Órdenes de cambios de ingeniería
13. Estándares y procedimientos de ingeniería del software

# Gestión de la Configuración del Software (GCS)

---

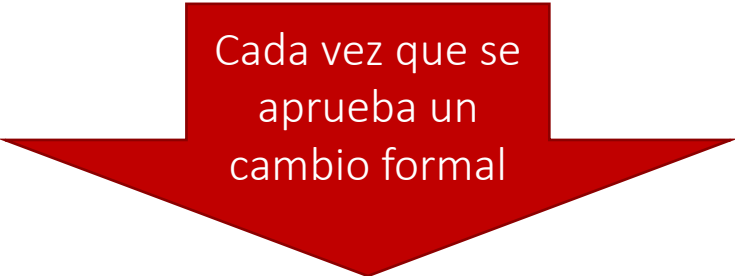
»El cambio se puede producir en cualquier momento, las actividades de la GCS sirven para:

Identificar el cambio

Controlar el cambio

Garantizar que el cambio se implemente adecuadamente

Informar del cambio a todos aquellos que puedan estar afectados



Cada vez que se  
aprueba un  
cambio formal

Línea Base

# Gestión de la Configuración del Software (GCS)

## »Línea Base

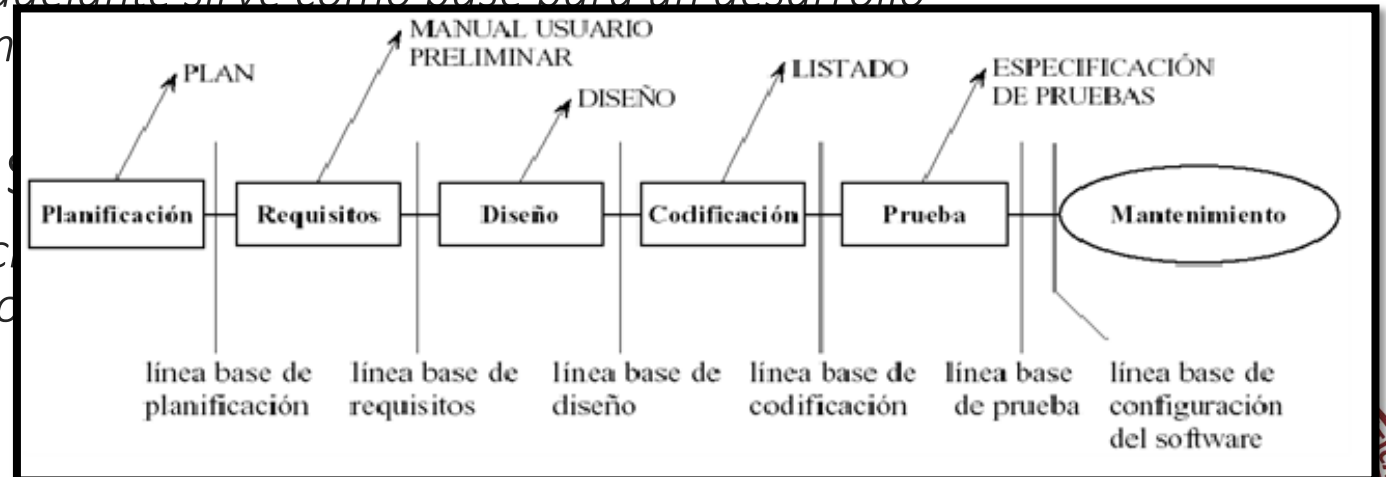
Una línea base es un concepto de GCS que nos ayuda a controlar los cambios

Definición de la IEEE

*Una especificación o producto que se ha revisado formalmente y sobre el 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 un control de cambio*

En el contexto de la Ingeniería de Software

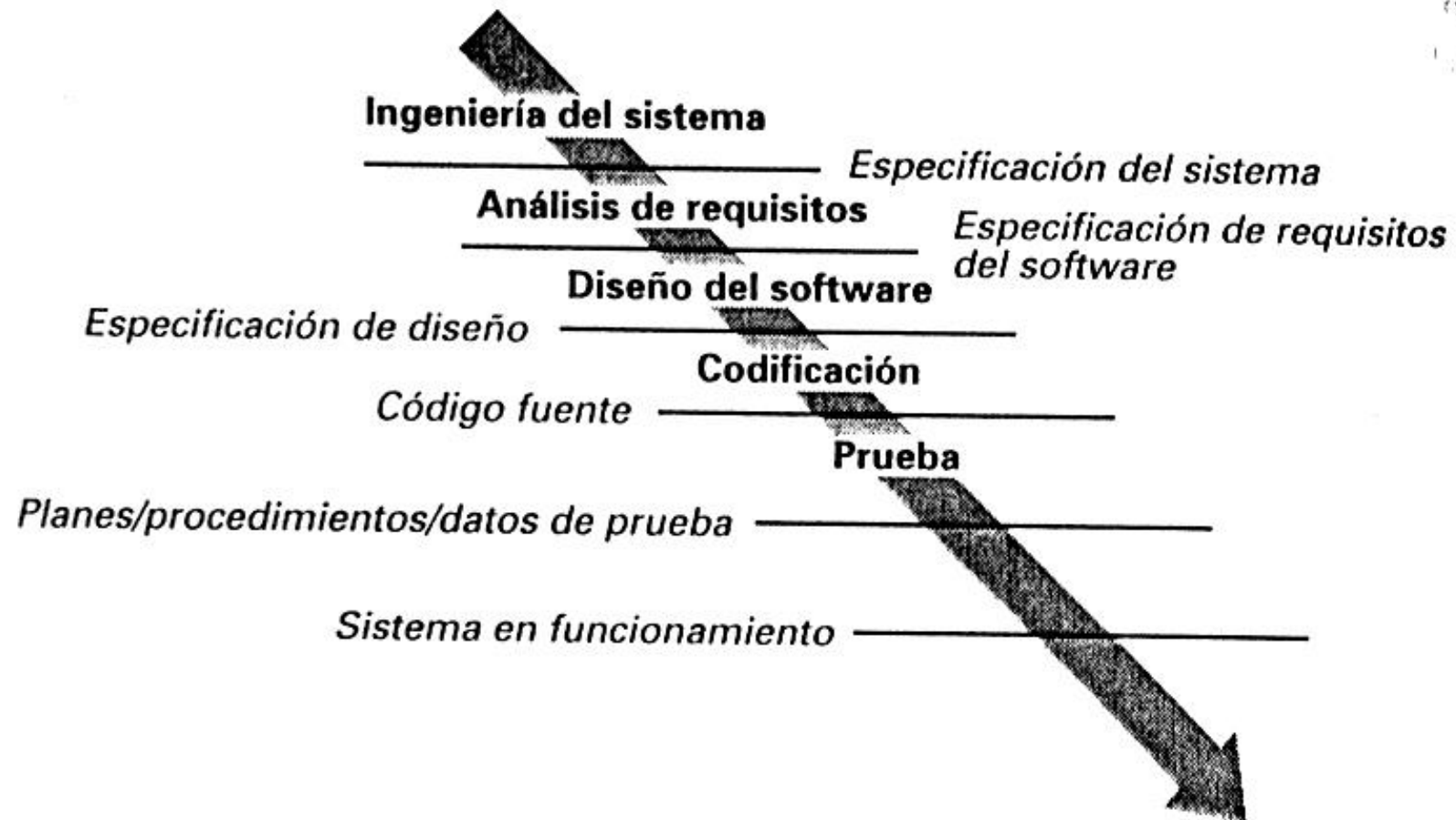
*Una línea base es un punto de referencia para el envío de uno o más ECS y su aprobación*





# Gestión de la Configuración del Software (GCS)

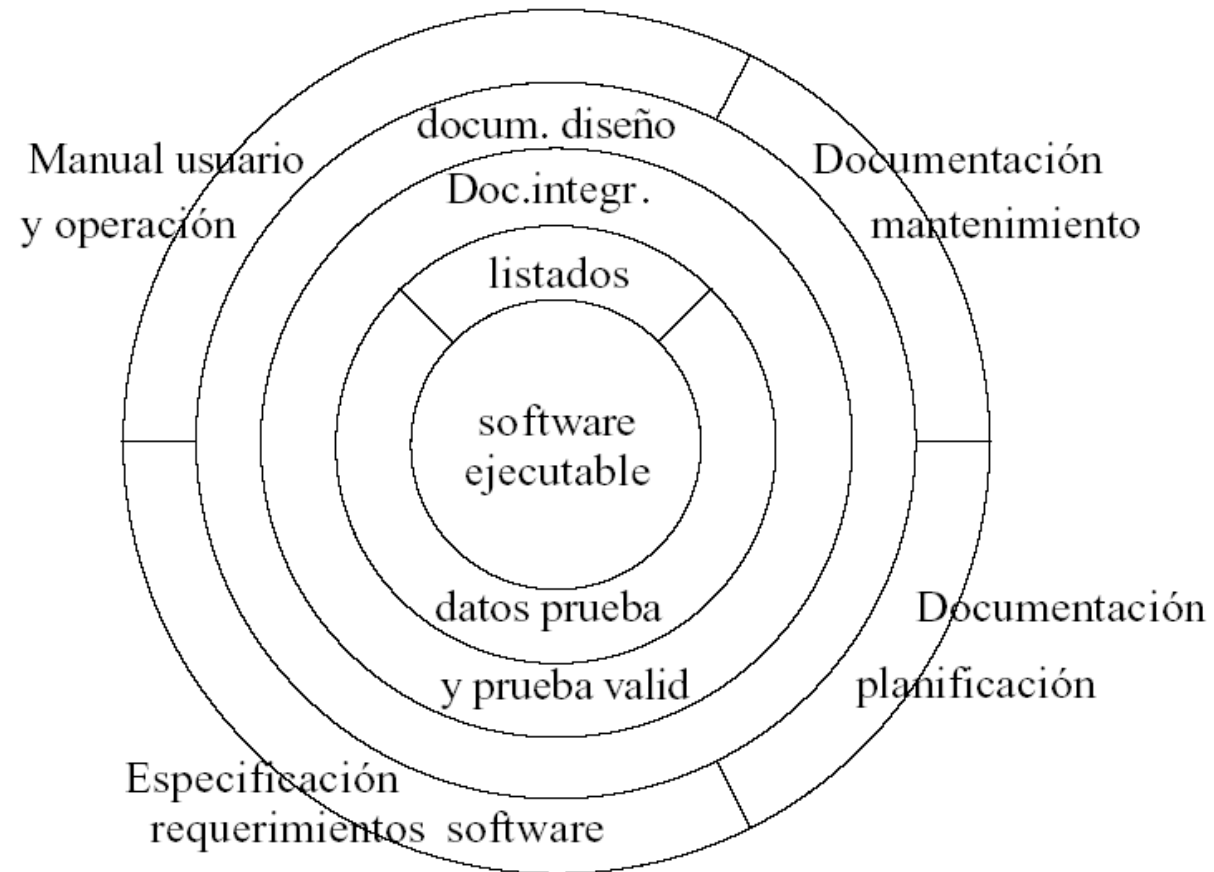
## » Línea Base



8



# Gestión de la Configuración del Software (GCS)



9

# Gestión de la Configuración del Software (GCS)

---

## »Importancia de la GCS

¿Cómo identifica y gestiona una organización las diferentes versiones existentes de un programa (y su documentación) de forma que se puedan introducir cambios eficientemente?

¿Cómo controla la organización los cambios antes y después de que el software sea distribuido al cliente?

¿Quién tiene la responsabilidad de aprobar y de asignar prioridades a los cambios?

¿Cómo podemos garantizar que los cambios se han llevado a cabo adecuadamente?

¿Qué mecanismo se usa para avisar a otros de los cambios realizados?

10



# Gestión de la Configuración del Software (GCS)

---

## »Proceso de la GCS

- Identificación
- Control de versiones
- Control de cambios
- Auditorías de la configuración
- Generación de informes

11



# Gestión de la Configuración del Software (GCS)

## »Proceso de la GCS

### 1- Identificación de los objetos en la GCS

*Nombre: cadena de caracteres sin ambigüedad*

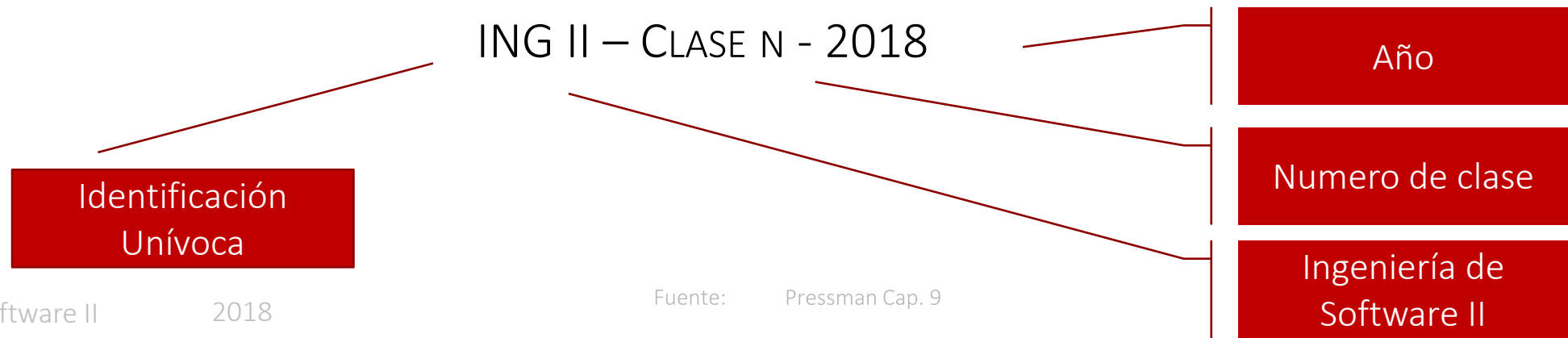
*Descripción: lista de elementos de datos que identifican:*

Tipo de ECS (documento, código fuente, datos)

Identificador del proyecto

Información de la versión y/o cambio

12



# Gestión de la Configuración del Software (GCS)

---

## »Proceso de la GCS

### 2 - Control de versiones

*Permite al usuario especificar configuraciones alternativas del sistema mediante la selección de versiones adecuadas ( por ejemplo asociando atributos que la identifican)*

*Combinación de procedimientos y herramientas para gestionar las versiones de los ECS*

*Ejemplo de versiones*

Un programa puede contener los módulos 1-2-3-4-5

Una versión puede utilizar los módulos 1-2-3-5

Otra versión puede utilizar los módulos 1-2-4-5

Dos variantes de una misma versión

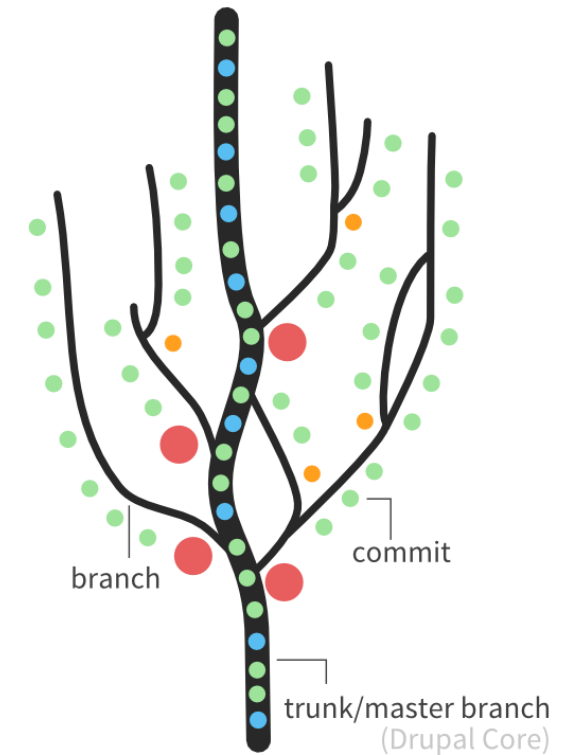
13



# Gestión de la Configuración del Software (GCS)

»Proceso de la GCS

## 2 - Control de versiones



14



# Gestión de la Configuración del Software (GCS)

---

## »Proceso de la GCS

### 2 - Control de versiones

#### *Repositorio*

Se almacenan los archivos actualizados e históricos de cambio del proyecto.

#### *Versión*

Determina un conjunto de archivos

#### *Master*

Conjunto de archivos principales del proyecto

#### *Abrir rama – branch*

Bifurcación del máster para trabajar sobre dos ramas de forma independiente

#### *Desplegar – check-out*

Copia de trabajo local desde el repositorio.

#### *Publicar - Commit*

Una copia de los cambios hechos a una copia local es escrita o integrada sobre repositorio.

#### *Conflicto*

Problema entre las versiones de un mismo documento

#### *Cambio – diff*

Representa una modificación específica

#### *Integración – Merge*

Fusión entre dos ramas del proyecto

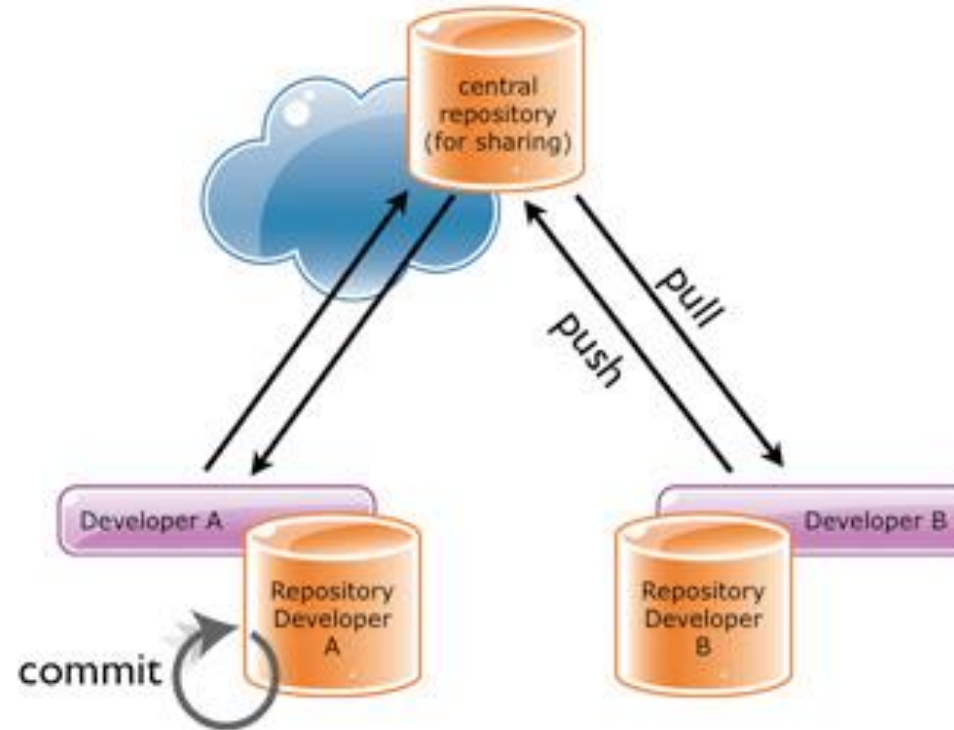
#### *Actualización – sync o update*

Integra los cambios que han sido hechos en el repositorio y las copias locales

# Gestión de la Configuración del Software (GCS)

## »Proceso de la GCS

### 2 - Control de versiones

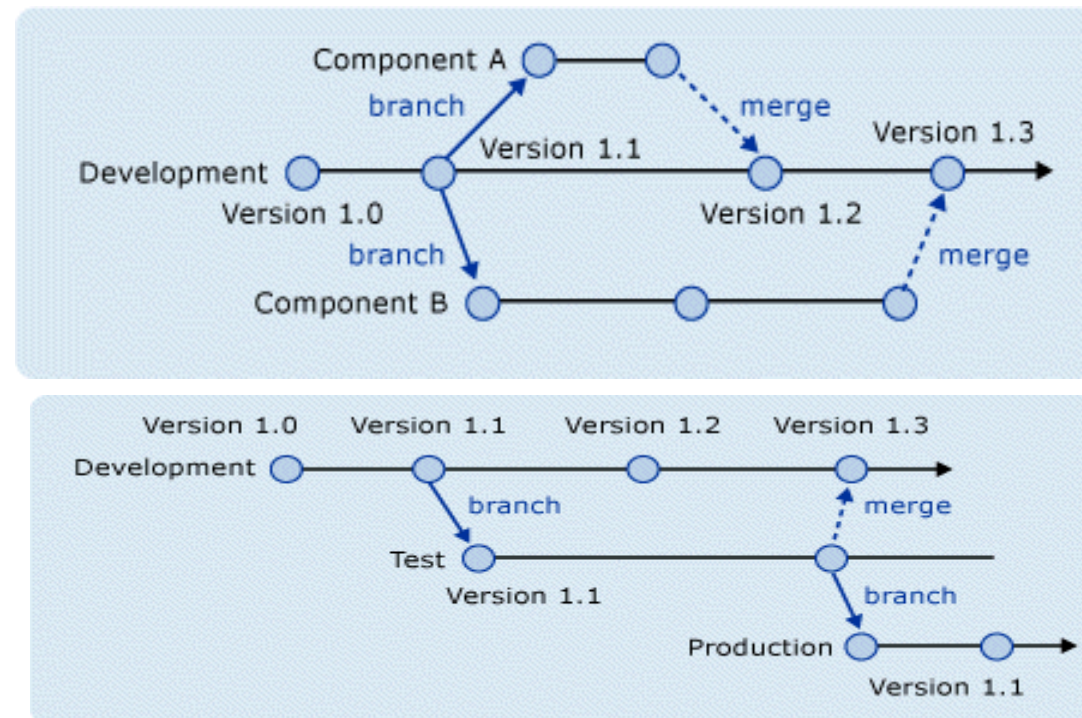


16

# Gestión de la Configuración del Software (GCS)

## »Proceso de la GCS

### 2 - Control de versiones



17

# Gestión de la Configuración del Software (GCS)

---

## »Proceso de la GCS

### 3 - Control de cambios

*A lo largo del proyecto los cambios son inevitables y el control es vital para el desarrollo del mismo*

*Combina los procedimientos humanos y las herramientas adecuadas para proporcionar un mecanismo para el control del cambio*

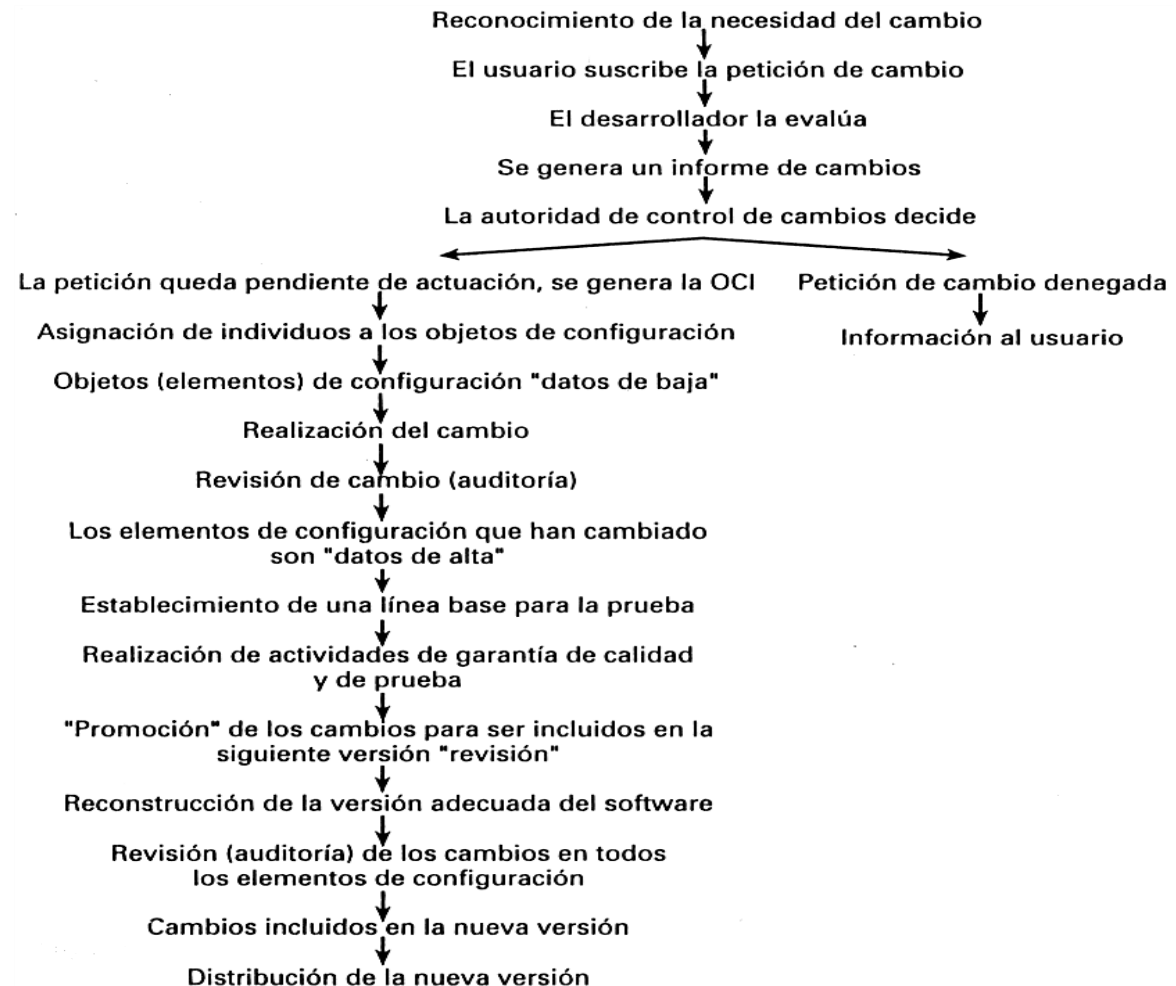
18



# Gestión de la Configuración del Software (GCS)

## »Proceso de la GCS

### 3 -Control de cambios



19



# Gestión de la Configuración del Software (GCS)

---

## »Proceso de la GCS

### 3 -Control de cambios

*La autoridad de control de cambios (ACC) evalúa:*

¿Cómo impactará el cambio en el hardware?

¿Cómo impactará el cambio en el rendimiento?

¿Cómo alterará el cambio la percepción del cliente sobre el producto?

¿Cómo afectará el cambio a la calidad y a la fiabilidad?

...

20





# Gestión de la Configuración del Software (GCS)

---

## »Proceso de la GCS

### 4 - Auditoría de la configuración

*La identificación y el control de versiones y el control de cambio, ayudan al equipo de desarrollo de software a mantener un orden, pero sólo se garantiza hasta que se ha generado la orden de cambio.*

*Cómo aseguramos que el cambio se ha realizado correctamente*

Revisiones técnicas formales

Auditorías de configuración

21



# Gestión de la Configuración del Software (GCS)

---

## »Proceso de la GCS

### 4 - Auditoría de la configuración responde:

*¿Se ha hecho el cambio especificado en la Orden de Cambio? ¿Se han incorporado modificaciones adicionales?*

*¿Se ha llevado a cabo una RTF para evaluar la corrección técnica?*

*¿Se han seguido adecuadamente los estándares de IS?*

*¿Se han reflejado los cambios en el ECS: fecha, autor, atributos?*

*¿Se han seguido procedimientos de GCS para señalar el cambio, registrarlo y divulgarlo?*

*¿Se han actualizado adecuadamente todos los ECS relacionados?*

22



# Gestión de la Configuración del Software (GCS)

---

## »Proceso de la GCS

### 5 - Generación de informes de estado de la configuración

*Responde*

¿Qué pasó?

¿Quién lo hizo?

¿Cuándo pasó?

¿Qué más se vio afectado?

*La generación de informes de estado de la configuración desempeña un papel vital en el éxito del proyecto*

23





# Ingeniería de software II

## Gestión De Proyectos



# ¿Qué es un proyecto ?

---

» Un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único.

» Características

Temporal

*Tiene un comienzo y fin definido. El fin se alcanza cuando se han logrado los objetivos del proyecto o cuando queda claro que esos objetivos no serán o no podrán ser alcanzados.*

Resultado

*Productos, servicios o resultados únicos*

Elaboración gradual

*Desarrollar en pasos e ir aumentando mediante incrementos*

25



# Aspectos de la Gestión de Proyecto de Software

## »Personal (RRHH)

Es el elemento más importante. El equipo de dirección del proyecto debe identificar a los interesados, determinar sus requisitos y expectativas y, gestionar su influencia en relación con los requisitos para asegurar un proyecto exitoso.

## »Producto

El producto de software es intangible. A veces es difícil ver el progreso del proyecto

Los proyectos pueden crear:

- Un producto o artículo producido, que es cuantificable, y que puede ser un elemento terminado o un componente
- La capacidad de prestar un servicio como, por ejemplo, las funciones del negocio que respaldan la producción o la distribución
- Un resultado

## »Procesos

Un proceso de software proporciona el marco de trabajo desde el cual se puede establecer un plan detallado para el desarrollo del software

## »Proyecto

Los proyectos deben ser planeados y controlados para manejar su complejidad

26



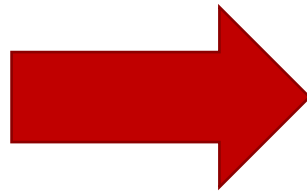


# Manifestación de una mala gestión de proyectos

---

- » Incumplimiento de plazos
- » Incremento de los costos
- » Entrega de productos de mala calidad

27



Perjuicio económico



# Elementos clave de la gestión de proyectos

---

- » Métricas
- » Estimaciones
- » Calendario temporal
- » Organización del personal
- » Análisis de riesgos
- » Seguimiento y control



PLANIFICACION

28

*La gestión de proyectos cubre todo el proceso de desarrollo*



# Ingeniería de software II

## Planificación

# Gestión De Proyectos – Planificación

---

## » Planificación

Planificación Temporal

Planificación Organizativa

30



# Planificación

---

» La planificación específica:

1. qué debe hacerse,
2. con qué recursos
3. y en qué orden;

Establece una secuencia operativa.

31



# Planificación

---

» ¿Por qué un software se retrasa?

- ✓ *Fecha límite de entrega poco realista*
- ✓ *Cambios en los requisitos*
- ✓ *Subestimación de los recursos necesarios*
- ✓ *Riesgos no considerados*
- ✓ *Dificultades humanas*
- ✓ *Falta de comunicación*

32





# Planificación

---

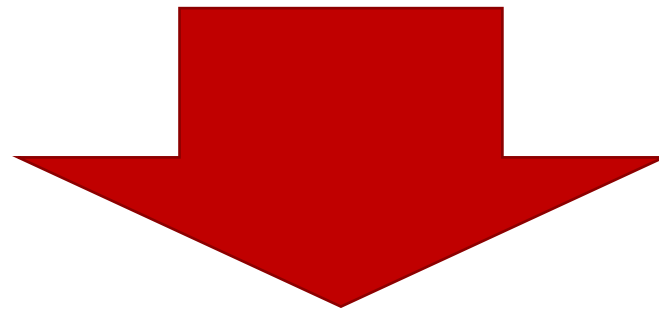
## » ¿Qué hacer?

Realizar una estimación detallada, determinando esfuerzo y duración.

Establecer la funcionalidad crítica.

Realizar reuniones con el cliente.

33



Planificar

# Gestión de Proyecto -

34

## Planificación Temporal

# Planificación temporal

---

Es una actividad que distribuye el esfuerzo estimado a lo largo de la duración prevista del proyecto

35

» La precisión de la planificación temporal es muy importante para no generar clientes insatisfechos, costos adicionales, reducción del impacto en el mercado, etc.



# Planificación Temporal

---

## »Perspectivas

Con fecha final establecida por el cliente

*Obligados a distribuir el esfuerzo dentro del plazo previsto*

Con fecha final fijada por los desarrolladores

*El esfuerzo se distribuye para conseguir un uso óptimo de los recursos y se define una fecha de fin luego de un cuidadoso análisis*

## »Lamentablemente la primera es la más frecuente

36



# Planificación Temporal

---

## »Principios Básicos

**Compartimentación:** dividir el proyecto en actividades y tareas manejables.

**Interdependencia:** determinar la interdependencia de cada actividad o tarea. Algunas pueden realizarse en paralelo, otras necesitan de la terminación de una anterior.

**Asignación de tiempo:** es decir, una fecha inicial y final, además de los recursos.

**Validación del esfuerzo:** Evita la sobreasignación.

**Responsabilidades definidas:** Asignación de responsables a cada tarea.

**Resultados definidos:** Cada tarea debe tener un resultado.

**Hitos definidos**

37



# Planificación Temporal

---

»Independientemente del modelo de proceso seleccionado, el modelo está compuesto de conjuntos de tareas o actividades.

Un Conjunto de tareas es una colección de tareas, hitos y entregas que se deben cumplir para completar el proyecto.

## *Tarea*

Secuencia de acciones a realizar en un plazo determinado.

## *Tarea Crítica*

Es aquella cuyo retraso genera un retraso en todo el proyecto.

## *Hito*

Es “algo” que se espera que esté hecho para alguna fecha, como por ejemplo, un módulo testeado o una característica del funcionamiento, un logro que sea objetivo, fácil de evaluar y notable.

## *Entregas*

Componente (fuentes, documentación, etc.) formalizados a través de un documento

38



# Planificación Temporal

---

## »Tareas

Una tarea puede describirse con cuatro parámetros :

***Precursor:** evento o conjunto de eventos que deben ocurrir antes de que la actividad pueda comenzar.*

***Duración:** cantidad de tiempo necesaria para completar la actividad.*

***Fecha de entrega:** fecha para la cual la actividad debe estar completada.*

***Resultado:** Hito o componente listo.*

39



# Planificación Temporal

---

## »Red de tareas

Es una representación gráfica del flujo de las tareas desde el inicio hasta el fin de un proyecto

En algunos casos los conjuntos de tareas permiten realizar algunas actividades en paralelo.

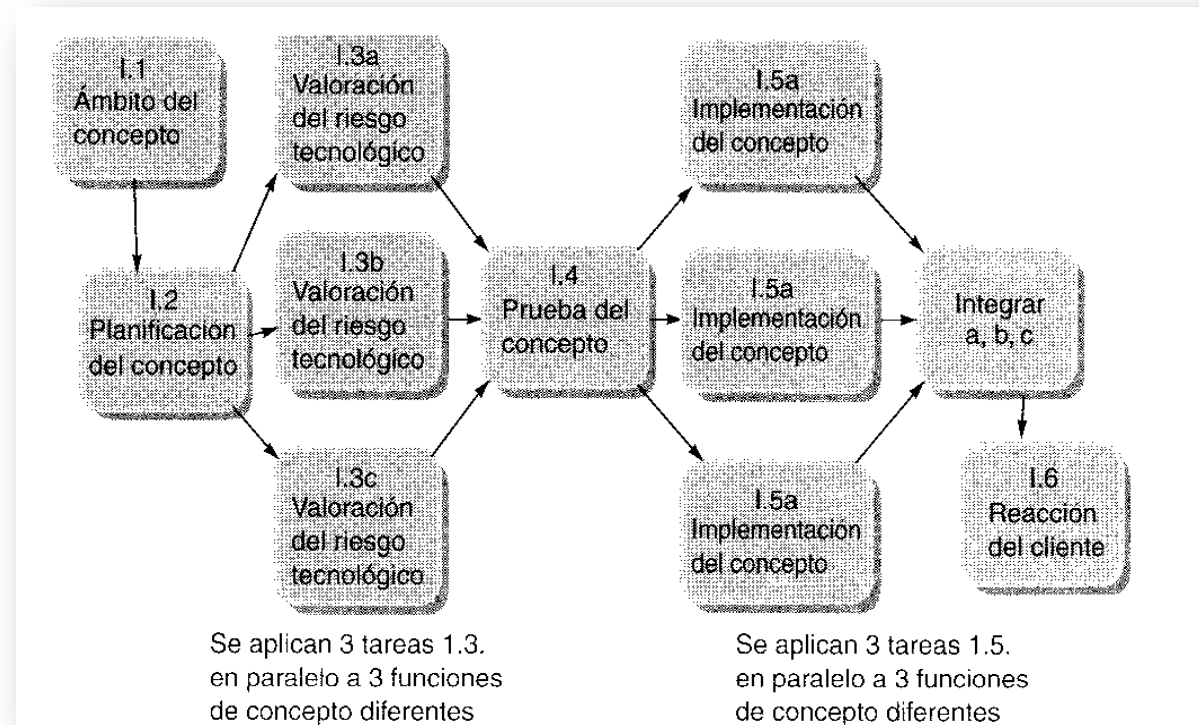
Representan la secuencia de las tareas y su interdependencia

40



# Planificación Temporal

## »Red de tareas



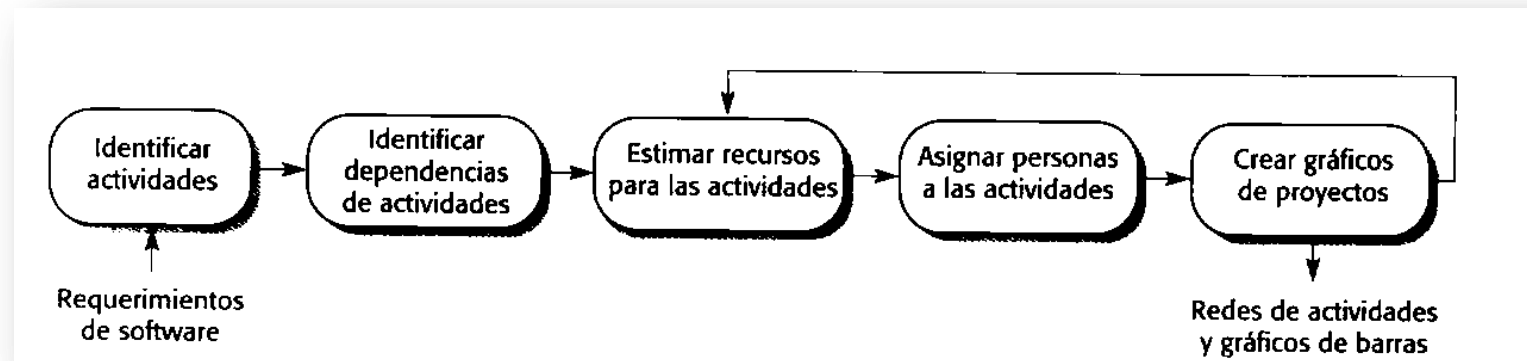
41

# Planificación Temporal

## »Calendarización

Separar todo el trabajo de un proyecto en actividades complementarias y considerar el tiempo requerido para completar dichas actividades

42



# Planificación Temporal

---

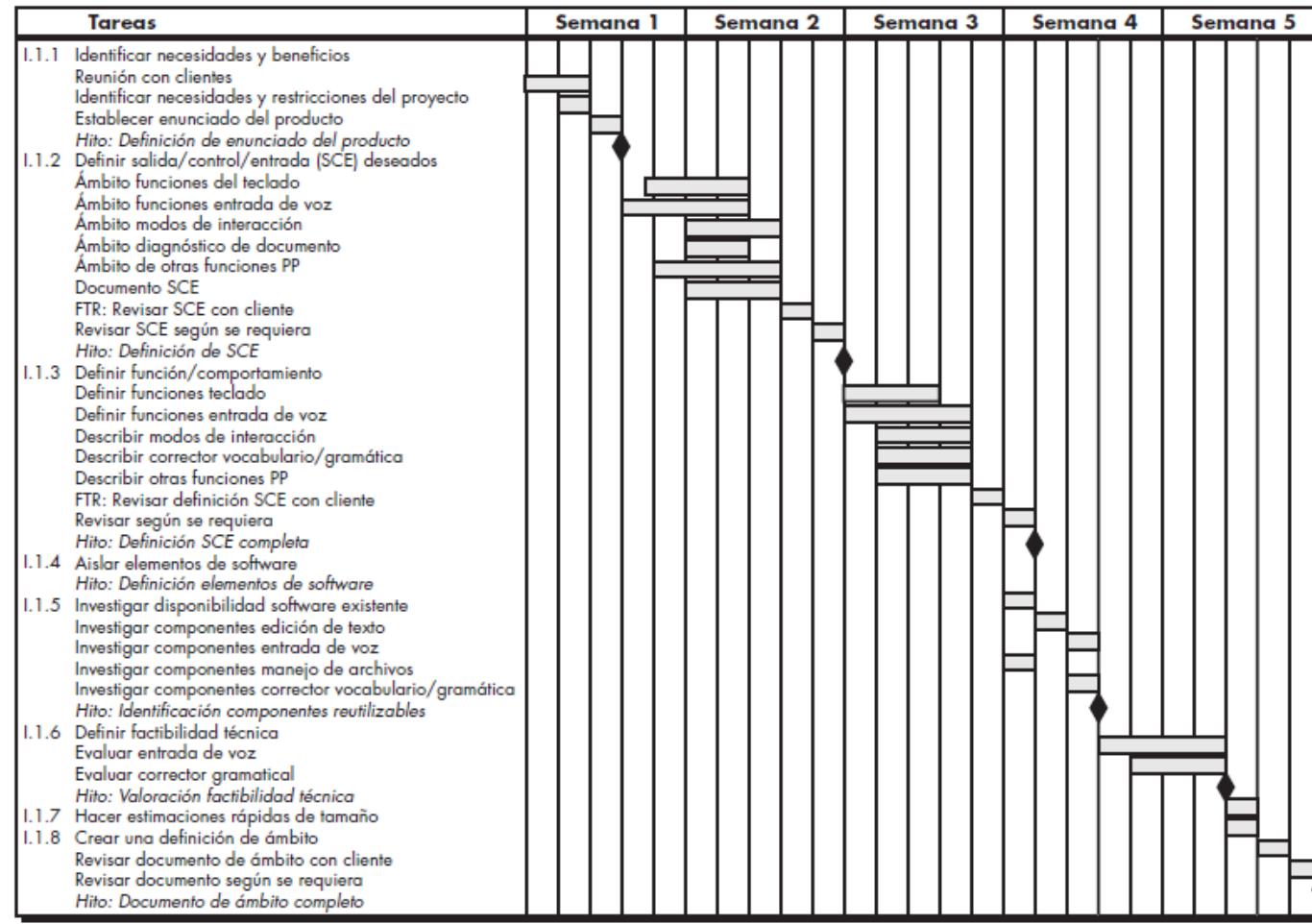
- » Cuando se crea un calendario de proyecto de software, se comienza con un conjunto de tareas.
- » Si se usan herramientas automatizadas, la distribución del trabajo se ingresa como una red de tareas. Luego se ingresan esfuerzo, duración y fecha de inicio para cada tarea. Además, las tareas pueden asignarse a individuos específicos.
- » Como consecuencia de esta entrada se genera un *cronograma*, también llamado *gráfico de Gantt*.

43



# Método de planificación temporal

## » GANTT



44



# Método de planificación temporal

---

## »PERT (Program Evaluation & Review Technique):

Creado para proyectos del programa de defensa del gobierno norteamericano entre 1958 y 1959.

Se utiliza para controlar la ejecución de proyectos con gran número de actividades que implican investigación, desarrollo y pruebas.

Red de tareas

*Fechas tempranas y tardías*

*Camino crítico*

## »CPM (Critical Path Method):

Desarrollado para dos empresas americanas entre 1956 y 1958.

Se utiliza en proyectos en los que hay poca incertidumbre en las estimaciones.

Tiempo de inicio temprano y tardío

45



# Método de planificación temporal

---

## »PERT y CPM

Actualmente se ha tomado lo mejor de ambos métodos y se han vuelto uno solo, conocido como Método del Camino Crítico.

*Establecer lista de tareas*

*Fijar dependencia entre tareas y duración*

*Construir la red*

*Numerar los nodos*

*Calcular la fecha temprana y tardía de cada nodo*

$T_{ei}$  = Fecha temprana del nodo  $i$

$T_{ai}$  = Fecha tardía del nodo  $i$

*Calcular el camino crítico que une las tareas críticas*

$\Rightarrow T_{ei} = T_{ai}$

46





# Método de planificación temporal PERT - CPM

---

## » Fechas Tempranas

$$TeJ = Tel + tIJ$$

Donde

*TeJ = fecha más temprana del nodo destino*

*Tel = fecha más temprana del nodo origen*

*tIJ = duración de la tarea desde el nodo I hasta el nodo J*

*Si hay más de un camino ... Max (TeJ1, TeJ2..)*

47

# Método de planificación temporal PERT - CPM

---

## » Fechas Tardías

$$TaI = TaJ - tIJ$$

Donde

*TaI = fecha más tardía del nodo origen*

*TaJ = fecha más tardía del nodo destino*

*tIJ = duración de la tarea desde el nodo I hasta el nodo J*

*Si hay más de un camino ... Min (TaJ1, TaJ2..)*

48



# Método de planificación temporal PERT - CPM

---

## »Margen Total

$$Mt = TaJ - Tel - tIJ$$

Donde

*TaJ = fecha tardía del nodo destino*

*Tel = fecha temprana del nodo origen*

*tIJ = duración de la tarea desde el nodo I hasta el nodo J*

*OBSERVAR que el Margen total también puede calcularse como  $MtJ = TaJ - Tel$*

*Es decir como la diferencia entre la fecha mas tardía y mas temprana del mismo nodo*

49



# Método de planificación temporal PERT - CPM

---

## » Método PERT - CPM

¿Qué ocurre cuando tengo un margen total de por ej. 6 días?

*Significa que la tarea puede iniciarse con 6 días de retraso sin que ello afecte a la duración total del proyecto.*

¿Qué ocurre cuando el margen total es 0?

*Significa que no hay margen y que esa tarea hay que iniciarla y finalizarla en las fechas más tempranas.*

*Puntualmente estas tareas con margen cero serían críticas.*

El camino formado por una sucesión de tareas críticas recibe el nombre de camino crítico.

El camino crítico puede obtenerse utilizando el cálculo del margen total.

50



# Método de planificación temporal PERT - CPM

---

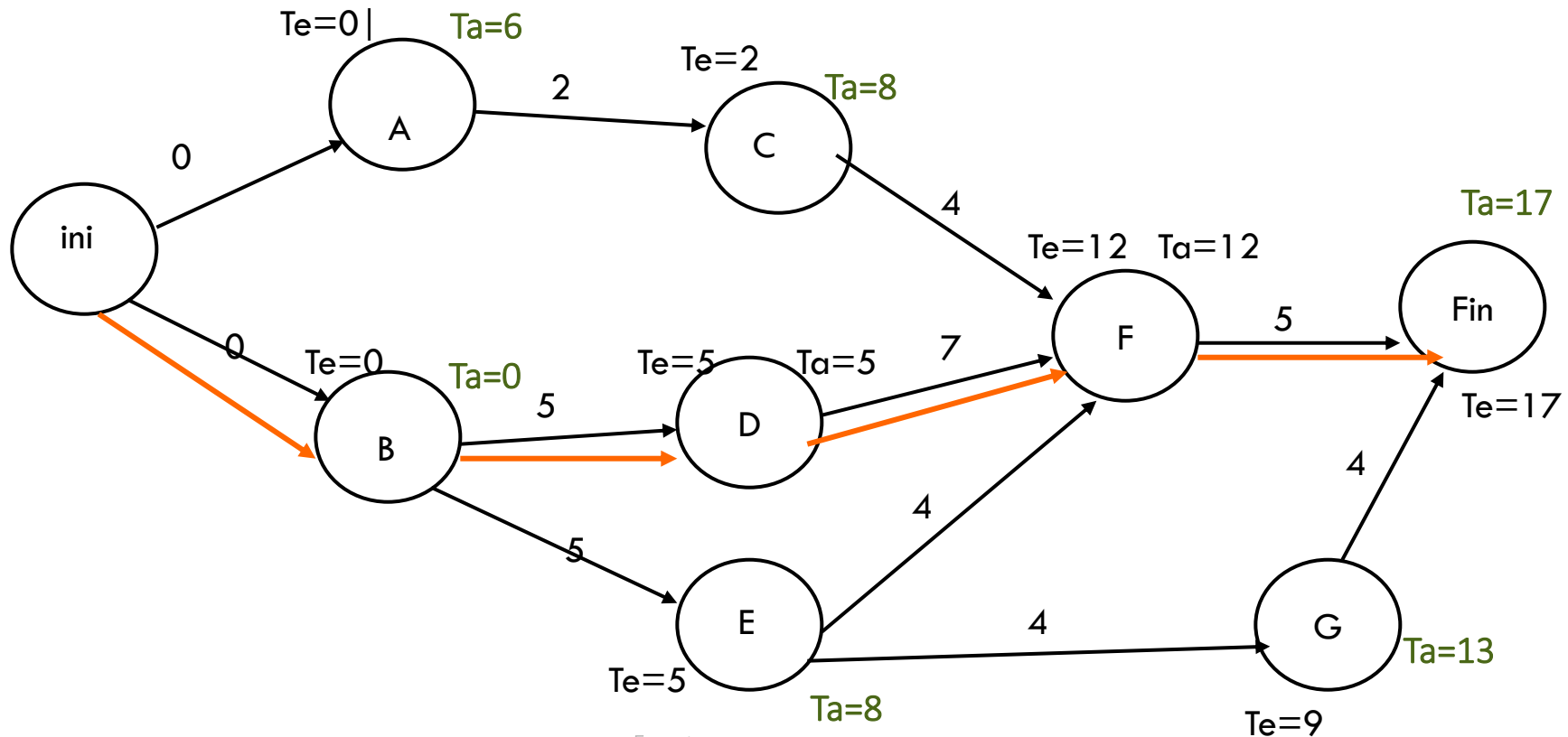
## »Ejemplo

Tarea	Precedida por	Duración
A	-	2
B	-	5
C	A	4
D	B	7
E	B	4
F	C-D-E	5
G	E	4

51



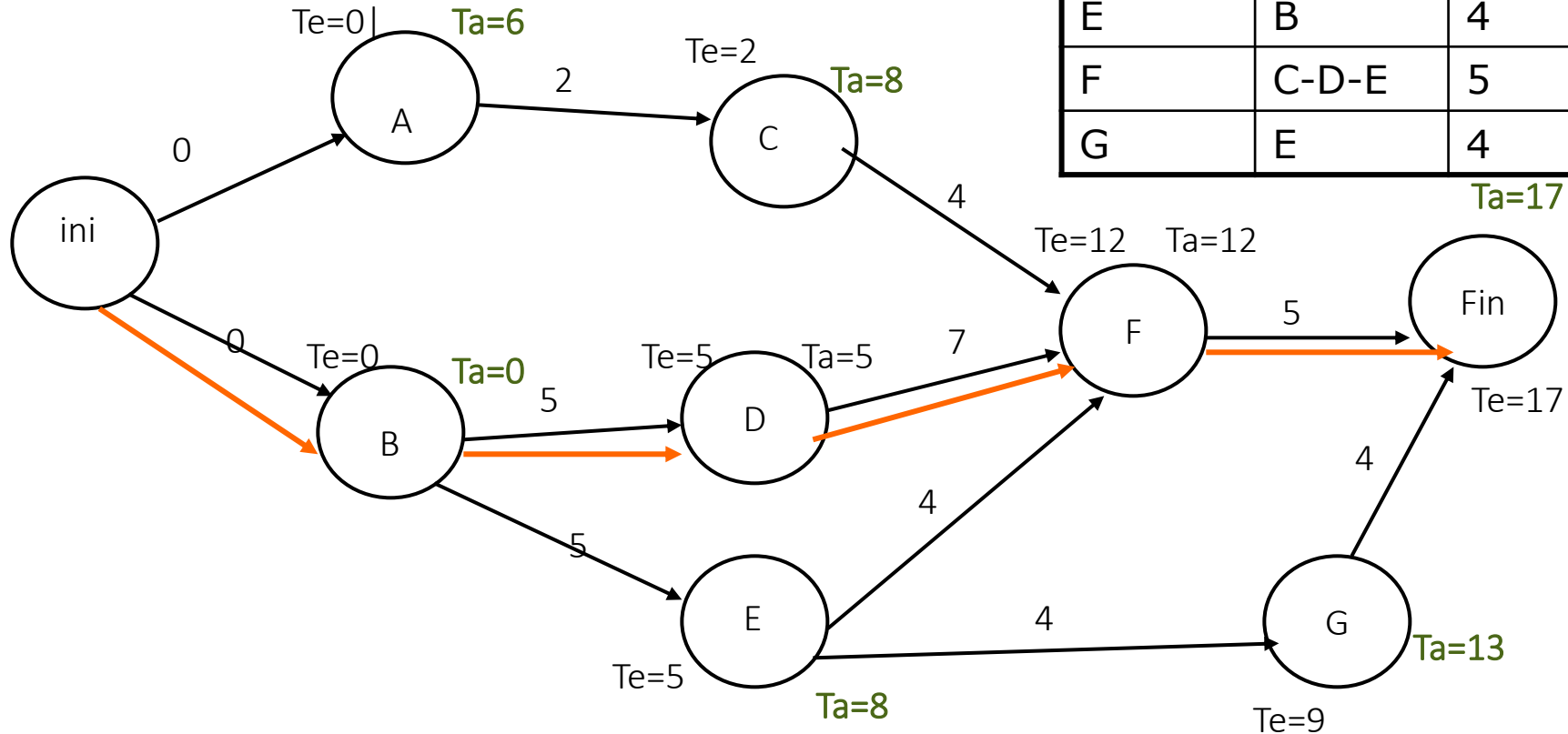
# Método de planificación temporal PERT - CPM



52

# Método de planificación temporal PERT - CPM

Tarea	Preced.	Dur.
A	-	2
B	-	5
C	A	4
D	B	7
E	B	4
F	C-D-E	5
G	E	4



53



# Método de planificación temporal

## PERT - CPM

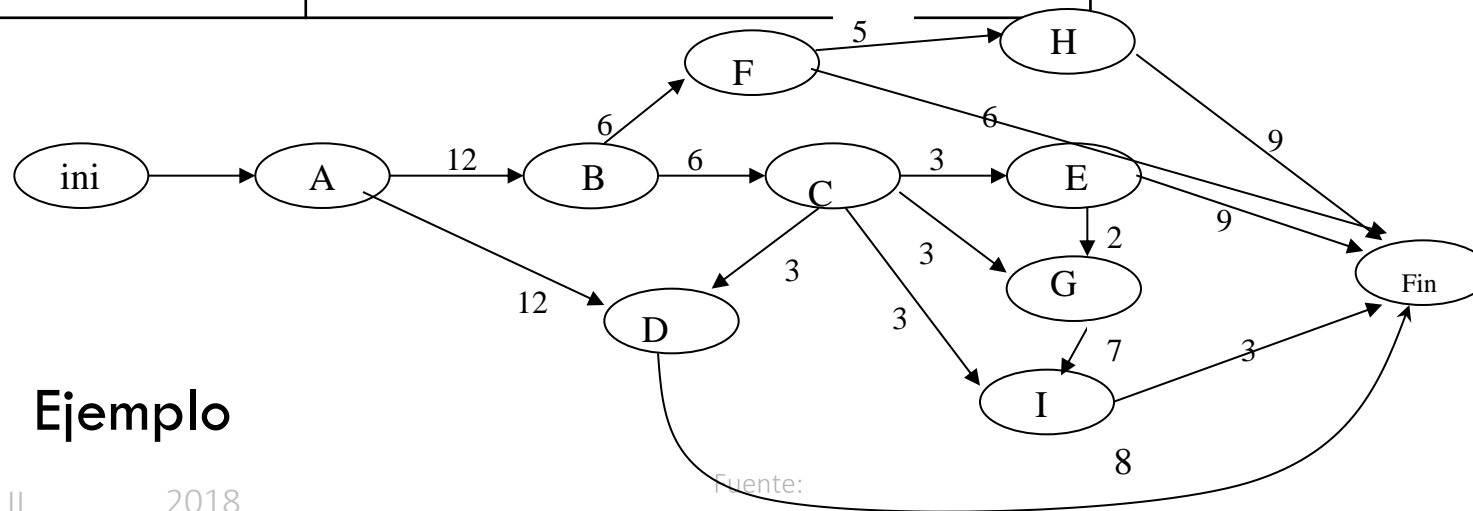
Tarea	Duración (semanas )	Restricciones
A	12	
B	5	A terminada
C	3	Empieza 1 semana después de terminada B
D	8	A terminada C terminada
E	9	C terminada
F	6	Empieza 6 semanas después del comienzo de B
G	4	C terminada. Empieza 2 semanas después del comienzo de E
H	9	Empieza 1 semana antes del fin de F
I	3	C terminada Empieza 3 semanas después del fin de G

54



# Método de planificación temporal PERT - CPM

Tarea	Duración (semanas )	Restricciones
A	12	
B	5	A terminada
C	3	Empieza 1 semana después de terminada B
D	8	A terminada C terminada
E	9	C terminada
F	6	Empieza 6 semanas después del comienzo de B
G	4	C terminada. Empieza 2 semanas después del comienzo de E
H	9	Empieza 1 semana antes del fin de F
I	3	C terminada Empieza 3 semanas después del fin de G

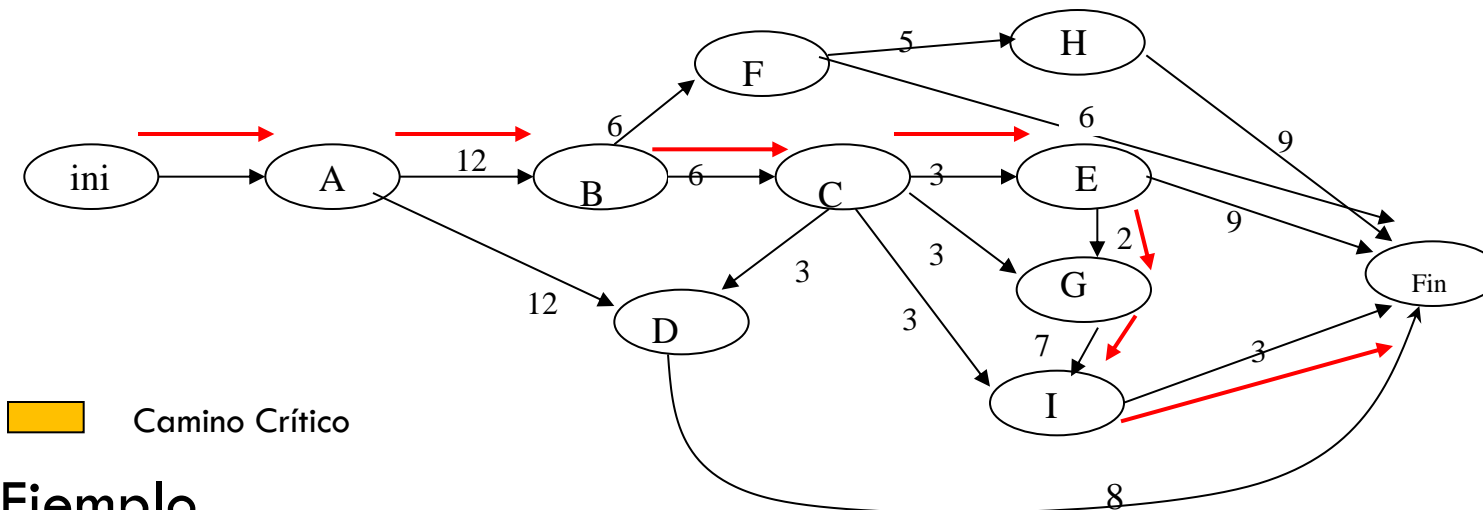


55



Tarea	Duración (semanas )	Restricciones	Te	Ta
A	12		0	0
B	5	A terminada	12	12
C	3	Empieza 1 semana después de terminada B	18	18
D	8	A terminada C terminada	21	25
E	9	C terminada	21	21
F	6	Empieza 6 semanas después del comienzo de B	18	19
G	4	C terminada. Empieza 2 semanas después del comienzo de E	23	23
H	9	Empieza 1 semana antes del fin de F	23	24
I	3	C terminada Empieza 3 semanas después del fin de G	30	30
Fin	0	-	33	33

56



Ejemplo

# Método de planificación temporal

---

» Datos que se obtienen del PERT - CPM:

Camino crítico

Ventana temporal para cada actividad

Fecha temprana de inicio de una tarea

Fecha tardía de inicio de una tarea sin retrasar la finalización del proyecto

Final más temprano de la tarea

Final más tardío de la tarea

Margen total

57

# Planificación temporal

---

¿QUÉ HACER CUANDO UNA TAREA SE SALE DE LA AGENDA?

Revisar el impacto sobre la fecha de entrega

Reasignar recursos

*La inclusión de más personas en el desarrollo no siempre genera aumento en la productividad*

Reordenar tareas

Modificar entrega

58



# Planificación temporal

---

## »Seguimiento y control del proyecto

Realizar reuniones periódicas para informar progresos y problemas

Evaluar las revisiones

Controlar que los “hitos” del proyecto se hayan alcanzado en la fecha programada

Comparar fecha estimada de inicio y real

Análisis del valor ganado (medida de progreso)

*Permite evaluar cuantitativamente el % de realización*

59



# Gestión de Proyecto

60

Planificación organizativa



# Planificación organizativa

---

- » El personal que trabaja en una organización de software es el activo más grande, representa el capital intelectual.
- » Una mala administración del personal es uno de los factores principales para el fracaso de los proyectos

61



# Planificación organizativa

---

## » Participantes

Gerentes ejecutivos

*Definen los temas empresariales*

Gerentes de proyecto

*Planifican, motivan, organizan y controla a los profesionales*

Profesionales

*Aportan habilidades técnicas*

Clientes

*Especifican los requerimientos*

Usuarios finales

*Interactúan con el software*

62



# Lideres de equipo

---

» ¿Que se busca cuando se busca de un líder de equipo?

» Modelo MOI de Liderazgo

**Motivación al personal** - Habilidades para alentar al personal técnico para producir a su máxima capacidad

**Organización del equipo** - Habilidades para modelar procesos que permitan que el concepto inicial se traduzca en un producto final

**Incentivación de Ideas e Innovación.** - Habilidad para alentar a las personas a crear y sentirse creativas.

» Rasgo de un líder eficaz

**Resolución de problemas** - Diagnosticar los conflictos técnicos y organizativos, estructurar una solución sistemática, aplica lecciones aprendidas, flexible

**Identidad administrativa** - Asumir el control cuando es necesario

**Logro** – recompensar las iniciativas

**Influencia y construcción del equipo** - Comprender señales verbales y no verbales y reaccionar ante las necesidades

63



# El equipo de software

---

»Factores a considerar cuando se planea una estructura de equipo

1. Dificultad de problema a resolver
2. Tamaño de programa resultante
3. Tiempo que el equipo permanecerá unido
4. Grado en el que puede dividirse en módulos el problema a
5. Calidad y confiabilidad requerida por el sistema a construir
6. Rigidez de la fecha de entrega
7. Grado de sociabilidad requerido en para el proyecto

64



# El equipo de software

---

## »Gestión de Grupos

Los grupos de desarrollo de software deberían ser pequeños y cohesivos.

### *Ventajas de los grupos cohesivos:*

- Los intereses del grupo son mas importantes que los personales
- Se pueden desarrollar estándares por consenso
- Se fomenta el aprendizaje de unos con otros
- Se garantiza la continuidad aún si un miembro abandona el equipo
- Los programas son una “propiedad” del grupo

### *Desventajas de los grupos cohesivos:*

- Resistencia al cambio por un liderazgo externo a los miembros del grupo
- Decisiones por mayoría sin estudiar alternativas

65



# El equipo de software

---

## »Comunicación Grupal

Las comunicaciones en un grupo se ven influenciadas por factores como: status de los miembros del grupo, tamaño del grupo, composición de hombres y mujeres, personalidades y canales de comunicación disponible.

66

Los programadores pueden mejorar la productividad si cuentan con un entorno de trabajo provisto con recursos necesarios y áreas de comunicación adecuadas.



# Planificación organizativa

---

## » Estructura de proyectos

### Proyecto

*Integración de un equipo de personas que llevan a cabo el proyecto de principio a fin.*

### Funcional

*Equipos distintos de personas que realizan cada fase del proyecto*

### Matricial

*Cada proyecto de desarrollo tiene un administrador. Cada persona trabaja en uno o más proyectos bajo la supervisión del administrador correspondiente*

67



# Planificación organizativa

---

## »Estructura del grupo

La “mejor” estructura de equipo depende del estilo de gestión de una organización, el número de personas que compondrá el equipo, sus niveles de preparación y la dificultad general del problema. Además depende de su naturaleza y del producto.

68





# Planificación organizativa

---

» Hay tres organigramas de equipos genéricos :

Descentralizado democrático (DD). Este equipo no tiene un jefe permanente. Se nombran coordinadores de tareas a corto plazo y se sustituyen por otros para diferentes tareas. Las decisiones se toman por consenso. La comunicación entre los miembros del equipo es horizontal.

69



# Planificación organizativa

---

» Hay tres organigramas de equipos genéricos :

Descentralizado controlado (DC). Este equipo tiene un jefe definido que coordina tareas específicas y jefes secundarios que tienen responsabilidades sobre subtareas. La resolución de problemas sigue siendo una actividad del grupo, pero la implementación de soluciones se reparte entre subgrupos por el jefe de equipo. La comunicación entre subgrupos e individuos es horizontal. También hay comunicación vertical a lo largo de la jerarquía de control.

70



# Planificación organizativa

---

» Hay tres organigramas de equipos genéricos :

Centralizado controlado (CC). El jefe del equipo se encarga de la resolución de problemas a alto nivel y la coordinación interna del equipo. La comunicación entre el jefe y los miembros del equipo es vertical.

71



# Planificación organizativa

---

## »Sugerencias

Una estructura centralizada (CC) realiza las tareas más rápidamente, es la más adecuada para manejar problemas sencillos.

Los equipos descentralizados generan más y mejores soluciones que los individuales, por tanto, estos equipos tienen más probabilidades de éxito en la resolución de problemas complejos.

La estructura DD es la mejor para problemas difíciles.

72



# Planificación organizativa

---

## »Sugerencias

Como el rendimiento de un equipo es inversamente proporcional a la cantidad de comunicación que se debe entablar, los proyectos muy grandes son mejor dirigidos por equipos con estructura CC o DC, donde se pueden formar fácilmente subgrupos.

El tiempo que los miembros del equipo vayan a «vivir juntos» afecta a la moral del equipo. Los equipos tipo DD producen una moral más alta y más satisfacción por el trabajo y son, por tanto, buenos para equipos que permanecerán juntos durante mucho tiempo.

73

