

Tema 1. Gestión de proyectos

- ✓ Importancia de la gestión
- ✓ Factores que influencian el éxito
- ✓ Personal
- ✓ Problema
- ✓ Proceso
- ✓ Modelos de proceso
- ✓ Actividades de gestión

Bibliografía

[SOM00] Ian Sommerville. "Software Engineering" (7^a ed.). Addison-Wesley. 2005

[PRES98] Roger S. Pressman. Ingeniería del sw. Un enfoque práctico (4^a ed.) Mc Graw-Hill. 1998

Capítulo 2: "El proceso"

Capítulo 3 "Conceptos sobre Gestión de proyectos"

[JAC99] El proceso unificado de desarrollo de software.

Capítulo 1: "El proceso Unificado: dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental".

[PIAT96] Análisis detallado de Aplicaciones informáticas. Mario G. Piattini

Capítulo 3: "Ciclo de vida software"

Importancia de la gestión

Buena gestión = condición **NECESARIA** para el
ÉXITO del proyecto

ES RESPONSABILIDAD DE LOS GESTORES...

- Planificar el proceso de desarrollo
- Hacer un seguimiento del trabajo, de forma que :
 - cumpla los **estándares** establecidos
 - se sigue la **agenda** prevista
 - no se sobrepasa el **presupuesto**

Conceptos sobre gestión (II)

Características especiales del software:

- Producto intangible
- No entendimiento claro sobre el proceso del sw
- Unicidad de los grandes sistemas software

Conceptos sobre gestión (III)

Claves para una **BUENA GESTIÓN**

Personal → Esfuerzo humano intenso

Problema → Minuciosa comunicación con cliente

Proceso → Métodos técnicos y herramientas

Personal

- Instituto de Ingeniería del Software crea “*Modelo de madurez de la capacidad de gestión del personal*”:
- Ayudando a atraer, aumentar, motivar, desplegar y retener el talento.
- Define las siguientes áreas clave:
 - reclutamiento,
 - selección,
 - gestión de rendimiento,
 - entrenamiento,
 - retribución,
 - desarrollo de la carrera,
 - diseño de la organización y del trabajo
 - desarrollo cultural y espíritu de equipo.

Personal

Participantes :

- ✓ Gestores Superiores
- ✓ Gestores Técnicos
- ✓ Profesionales
- ✓ Clientes
- ✓ Usuarios Finales



- Planificar
- Motivar
- Organizar y
- Controlar a los Profesionales

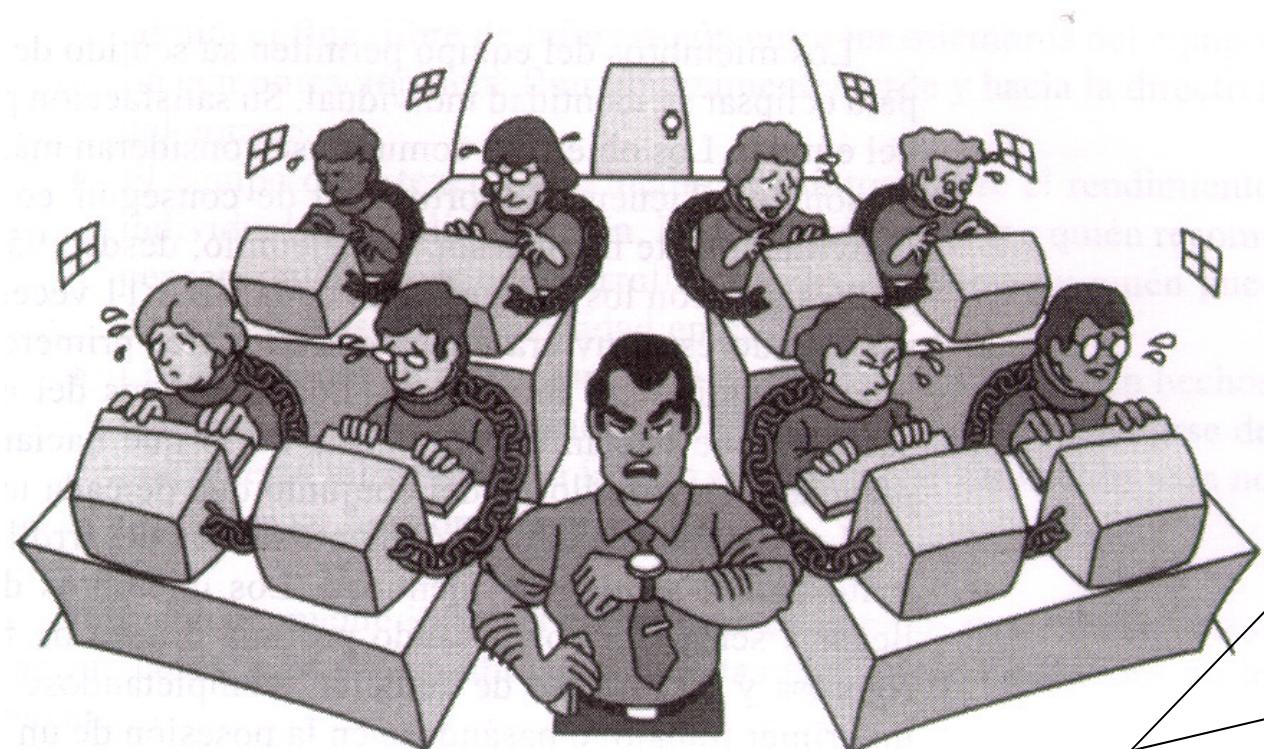
Personal (el jefe de equipo) (I)

Características :

- ✓ Habilidad para motivar
- ✓ Habilidad para moldear procesos (resol. problemas)
- ✓ Habilidad para incentivar la creatividad
- ✓ Dotes de gestión
- ✓ Saber incentivar los logros (increment. productividad)
- ✓ Capacidad para crear un equipo que presente cohesión

Personal (el jefe de equipo) (II)

La forma en que se presente el proyecto determinará si el equipo lo ve como una misión a realizar o como trabajos forzados



“¿Sois un equipo o no?
¿Por qué vamos tan despacio?
Trabajad más duro.”

Personal (el jefe de equipo) (III)



Desarrollador de software bien estimado

Personal (el equipo de software) (I)

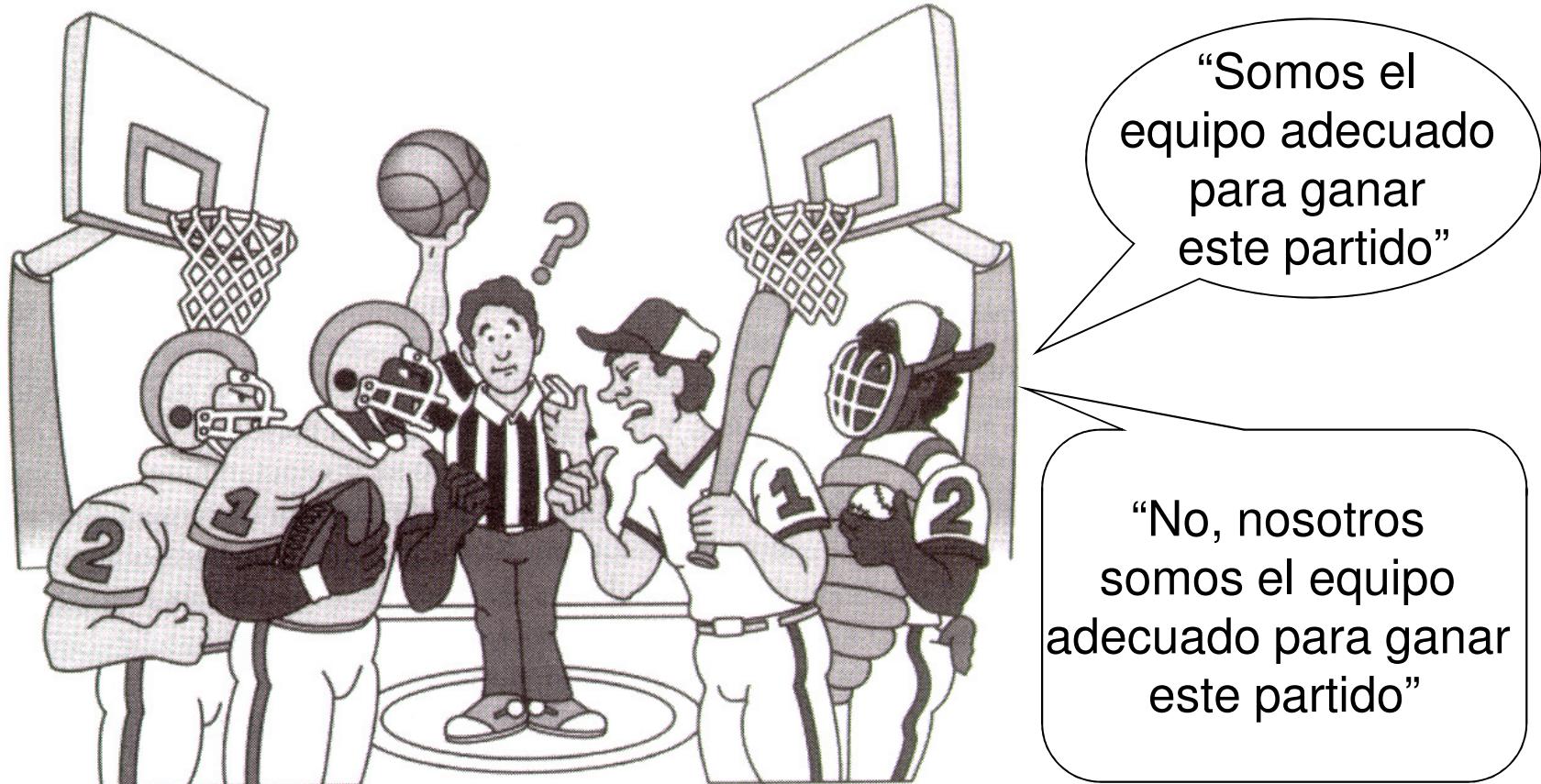
Organización del personal :

- ✓ **N** individuos asignados a **m** tareas funcionales ($m \geq N$)
- ✓ **N** individuos asignados a **m** tareas funcionales ($m < N$)
- ✓ **N** individuos organizados en **t** equipos

Estructuras de equipo :

- ✓ Descentralizado democrático (DD)
- ✓ Descentralizado controlado (DC)
- ✓ Centralizado controlado (CC)

Personal (el equipo de software) (II)



No hay una estructura única de equipo mejor para todos los proyectos

Personal (el equipo de software) (III)

El rendimiento de un equipo es INVERSAMENTE proporcional a la cantidad de comunicación que se deba establecer

El tiempo que los miembros del equipo vayan a “vivir juntos” afecta a la moral del equipo

Factores útiles para seleccionar personal son: experiencia en el dominio, adaptabilidad y personalidad.

Personal (el equipo de software) (IV)

Factores a tener en cuenta:

- ✓ Dificultad y tamaño del problema a resolver
- ✓ Tiempo de vida del grupo
- ✓ Grado de modularización del problema
- ✓ Calidad y fiabilidad requerida del sistema
- ✓ Rrigidez del tiempo de entrega
- ✓ Grado de comunicación requerida por el proyecto

Personal (el equipo de software) (VI)

		DD	DC	CC
DIFICULTAD	ALTA	X		
	PEQUEÑA		X	X
TAMAÑO	GRANDE		X	X
	PEQUEÑO	X		
DURACIÓN DEL EQUIPO			X	X
	CORTO			
	LARGO	X		
MODULARIDAD	ALTA		X	X
	BAJA	X		
FIABILIDAD	ALTA	X	X	
	BAJA			X
FECHA DE ENTREGA				X
	EXTRICTA			
	FLEXIBLE	X	X	
COMUNICACIÓN				
	ALTA	X		
	PEQUEÑA		X	X

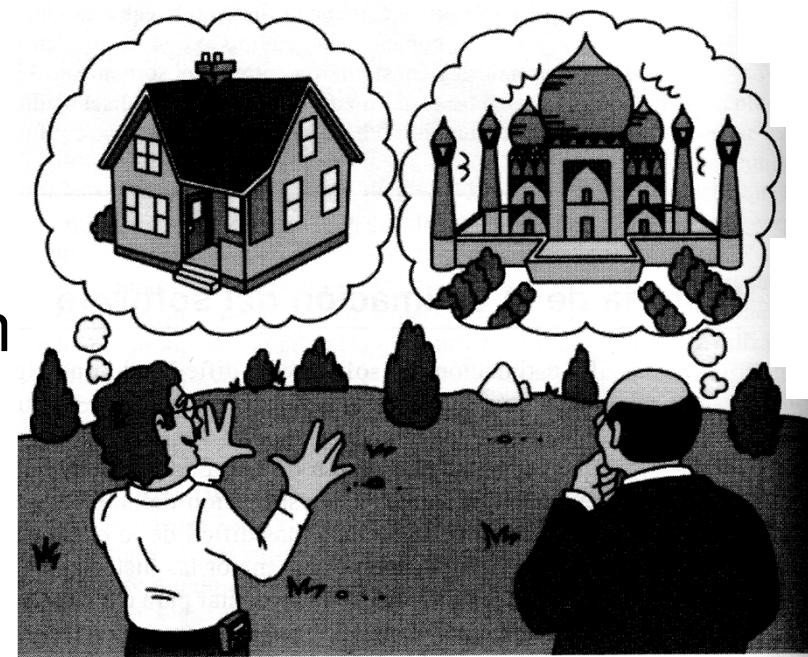
Problema

Para planificar un proyecto se requieren estimaciones
CUANTITATIVAS

Pasos:

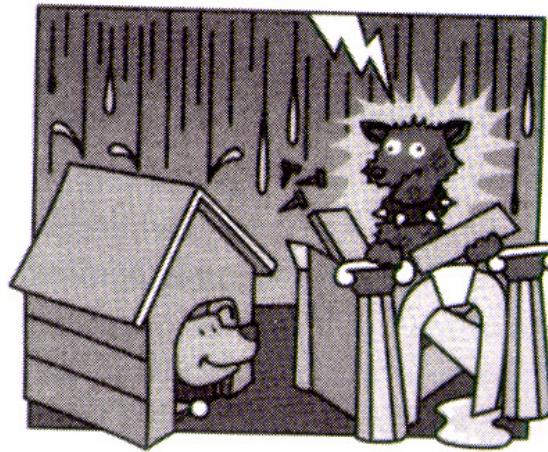
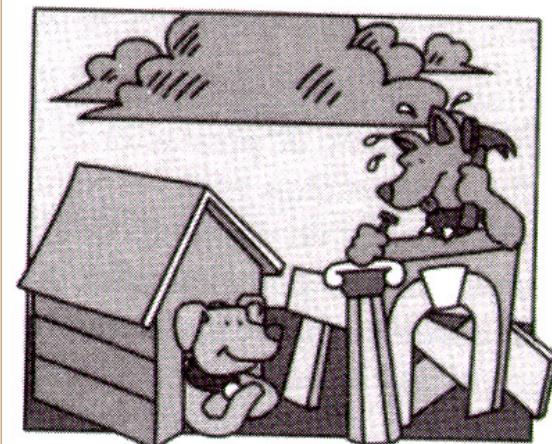
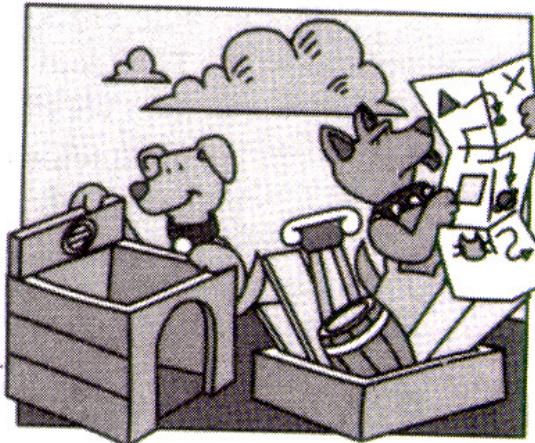
- ✓ Ámbito del software
 - Contexto
 - Objetivos de información
 - Función y rendimiento

- ✓ Descomposición del problema
 - Funcionalidad
 - Proceso



Proceso

Se debe seleccionar el modelo de proceso apropiado

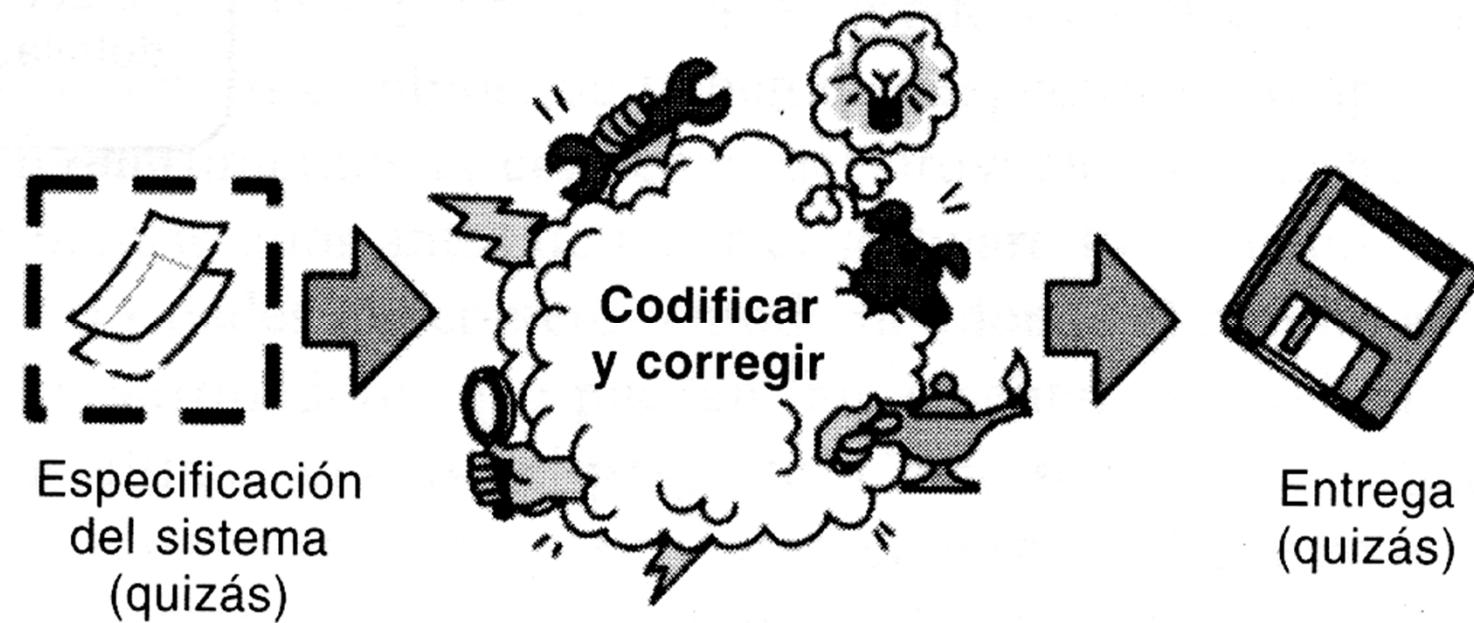


para la ingeniería
del software que
debe aplicar el
equipo del proyecto

Los proyectos
pequeños
necesitan
menos tiempo
para su
desarrollo

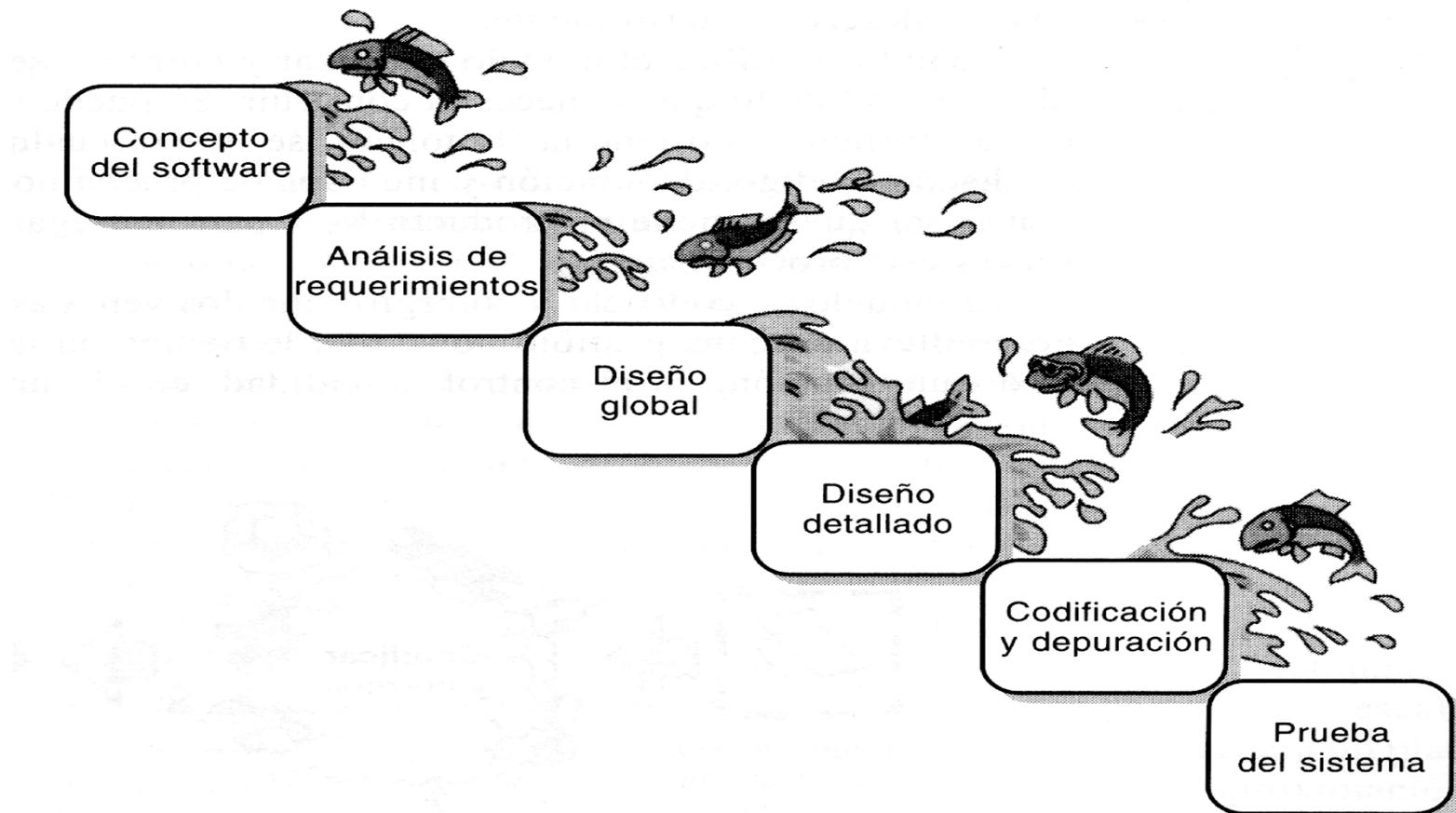
Modelos de Proceso

MODELO CODIFICAR Y CORREGIR



Modelos de Proceso Secuenciales (I)

MODELO DE CICLO DE VIDA EN CASCADA



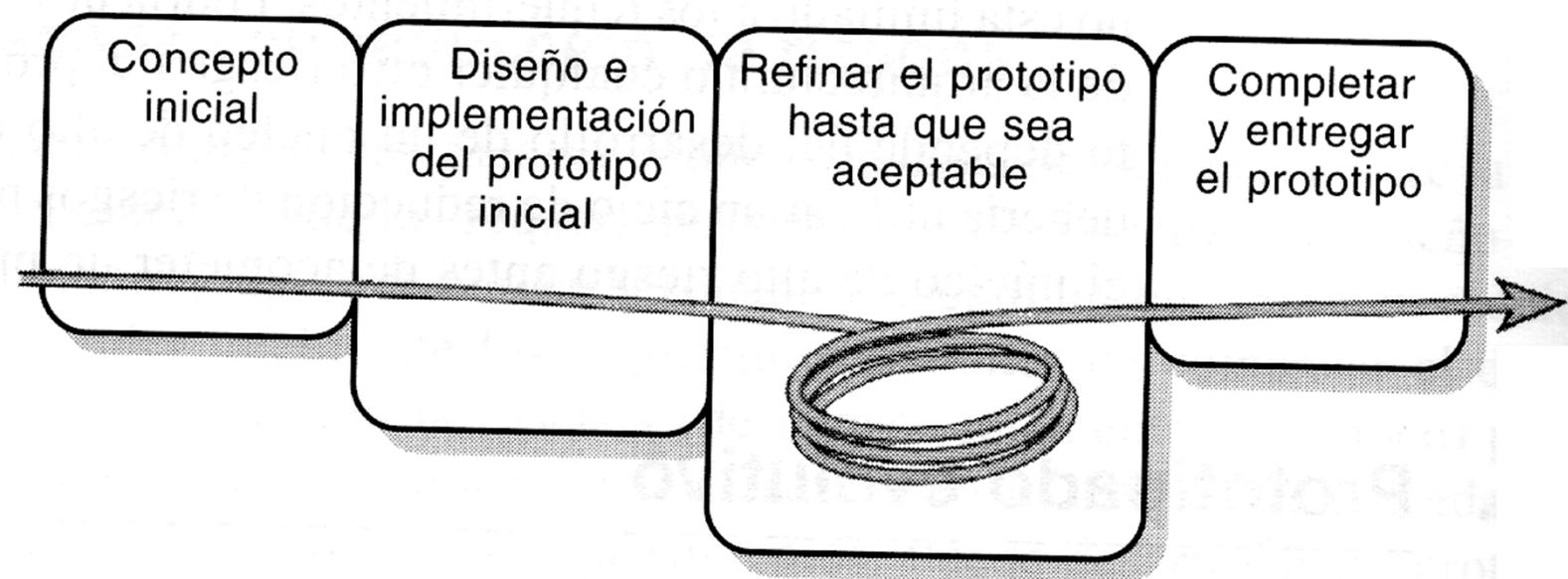
Modelos de Proceso Secuenciales (II)

CRÍTICAS AL MODELO DE CICLO DE VIDA EN CASCADA

- Acentúa el fracaso de la industria software frente al usuario final.
- Se tarda mucho tiempo en pasar por todo el ciclo, dado que hasta que no se finalice una fase no se pasa a la siguiente.
- No refleja el proceso real de desarrollo software. Los proyectos reales raramente siguen este flujo secuencial , puesto que siempre hay iteraciones.

Modelos de Proceso Secuenciales (III)

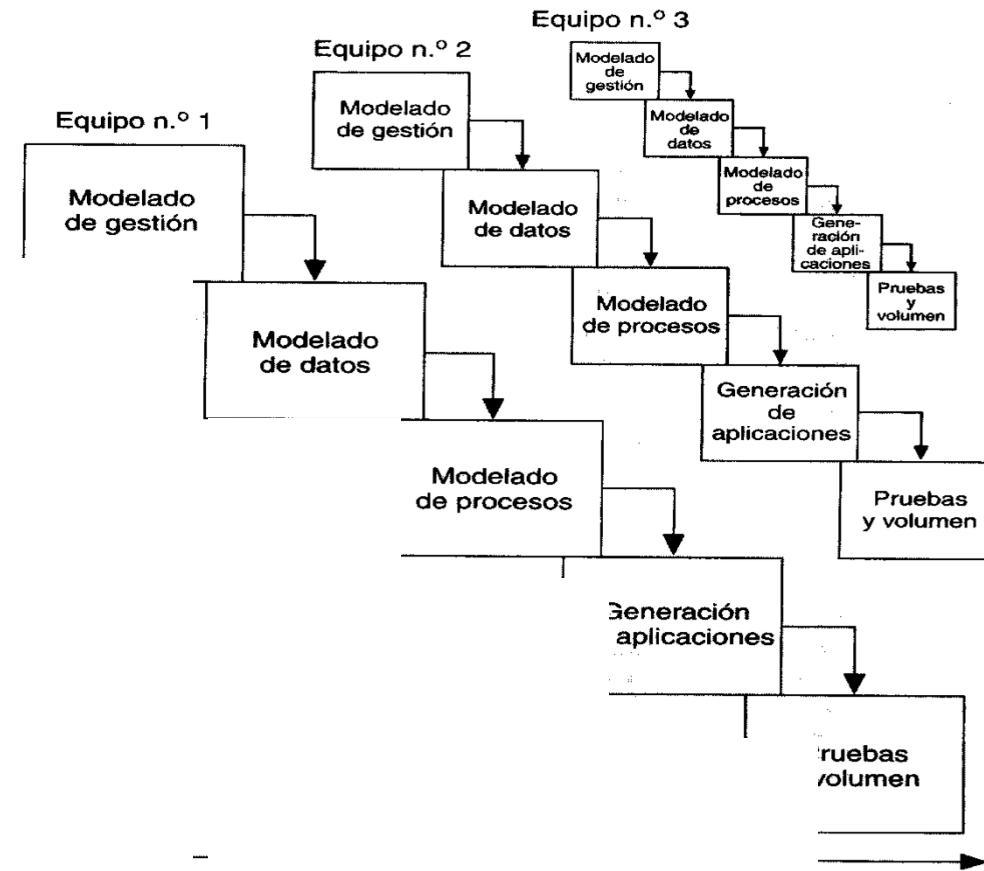
MODELO CONSTRUCCION DE PROTOTIPOS



Modelos de Proceso Secuenciales (VI)

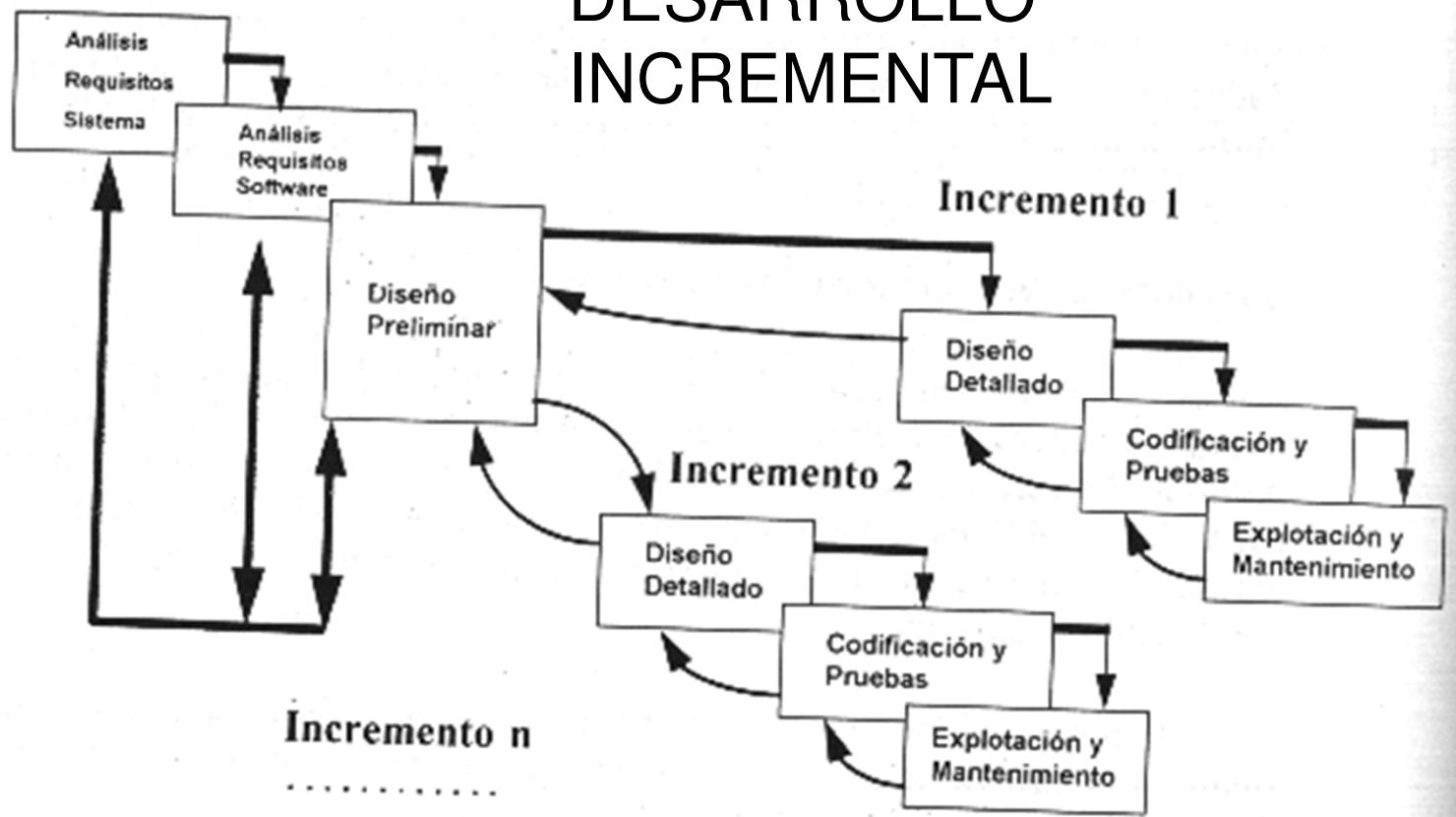
MODELO DE DESARROLLO RAPIDO DE APLICACIONES (DRA)

El modelo DRA



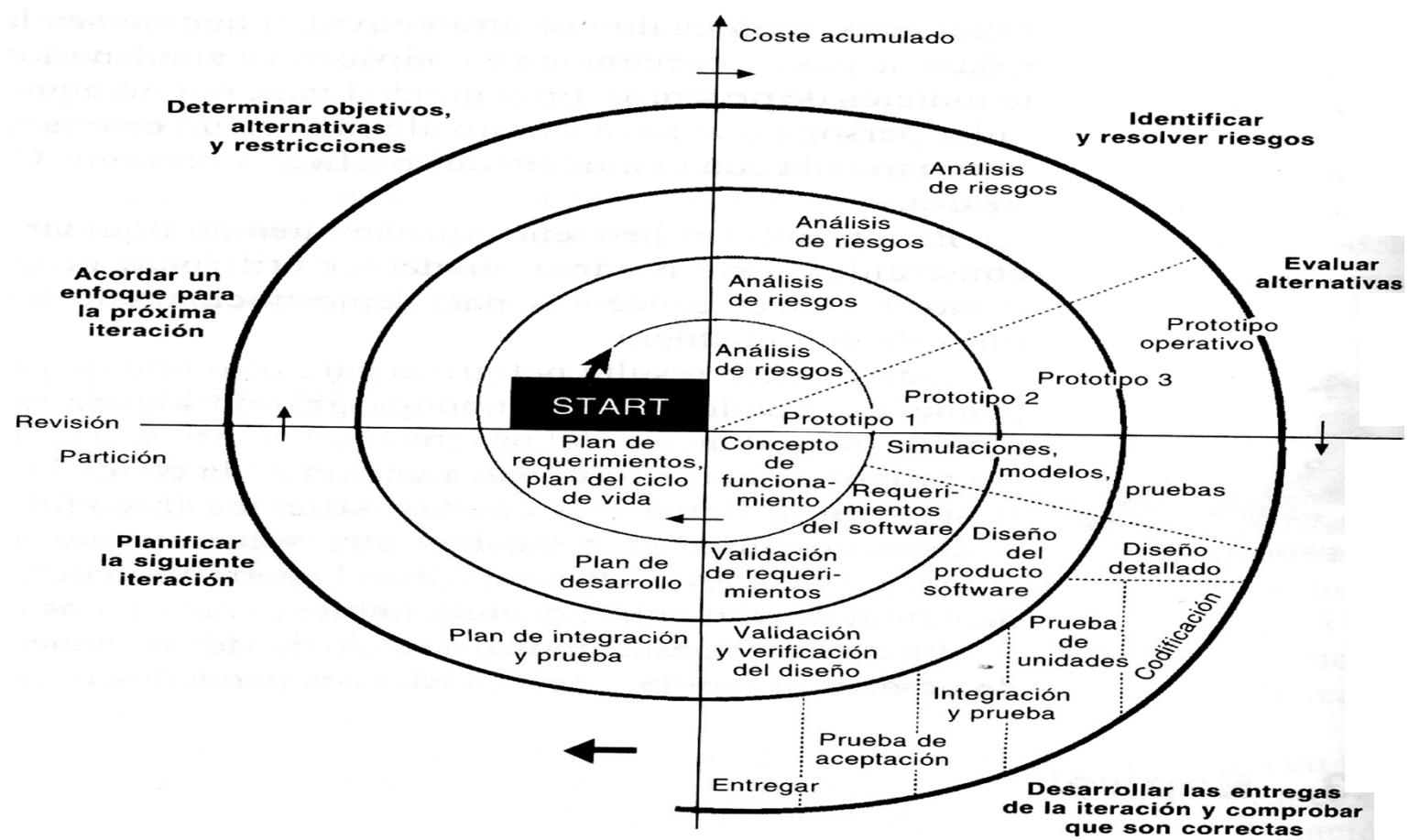
Modelos de Proceso Evolutivos (I)

MODELO DE DESARROLLO INCREMENTAL



Modelos de Proceso Evolutivos (II)

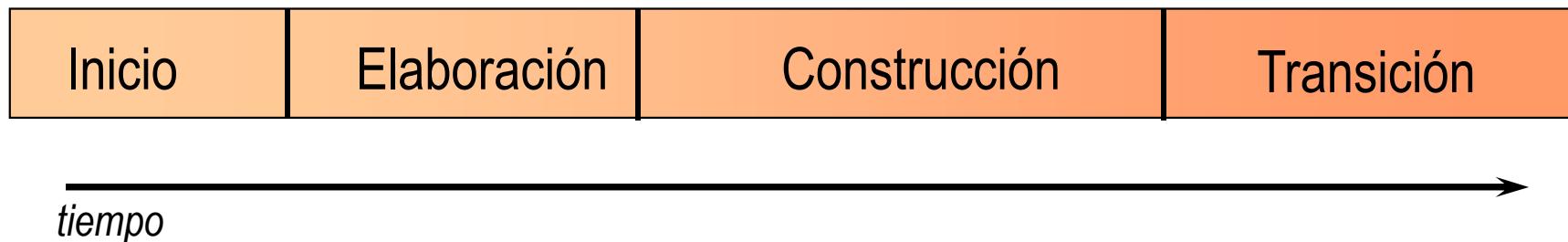
MODELO EN ESPIRAL



Modelos de Proceso Evolutivos

(III). (cont.)

DESARROLLO UNIFICADO

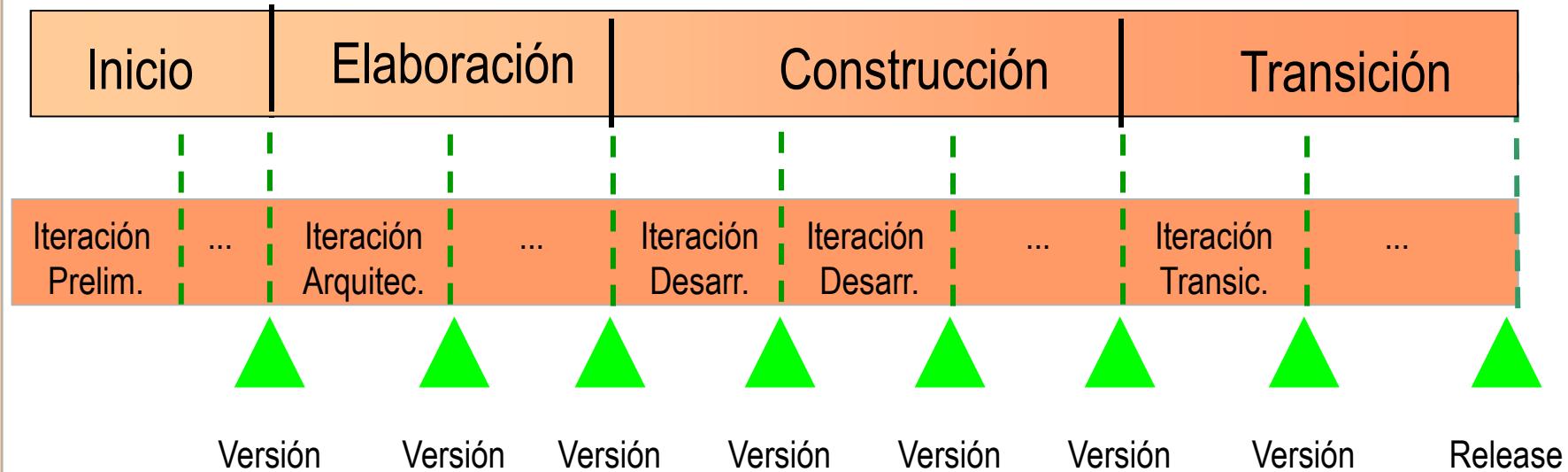


- **Inicio:** Se define el ámbito del proyecto y se desarrollan los casos de uso
- **Elaboración:** Se realiza el plan del proyecto, estimaciones, diseño básico
- **Construcción:** Se implementa en base a iteraciones
- **Transición:** Fase de transición para entregar el producto a los usuarios (p.ej. pruebas beta)

Modelos de Proceso Evolutivos

(III) (cont.)

DESARROLLO UNIFICADO



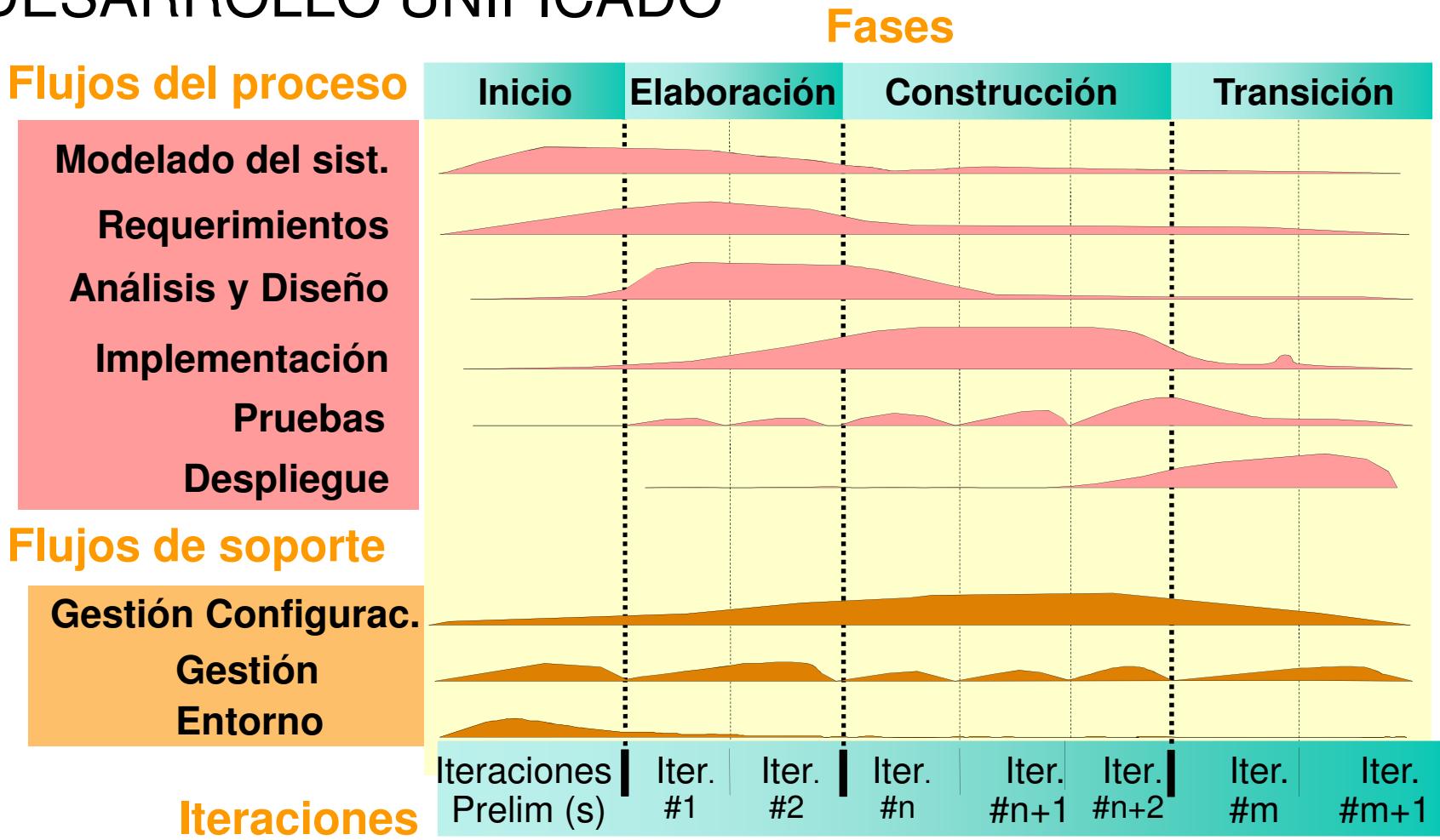
Una iteración es una secuencia de actividades con un plan establecido y un criterio de evaluación, que resulta en una *versión interna nueva (incremento)*

Al final de un ciclo obtenemos una versión para el cliente (release).

Modelos de Proceso Evolutivos

(III) (cont.)

DESARROLLO UNIFICADO



Modelos de Proceso Evolutivos (III)

DESARROLLO UNIFICADO: Conceptos clave

- Fases, Iteraciones
- Flujos del proceso
 - Actividades, pasos
- Artefactos
 - modelos
 - Informes, documentos
- Trabajador: Ingeniero

¿Cuándo tienen lugar?

¿Qué hay que hacer?

¿Qué se produce?

¿Quién lo hace?

Problema y proceso

Actividades del proceso	comunicación cliente	planificación	análisis de riesgos	ingeniería	
Tareas					
Funciones del producto					
Función 1					
Función 2					
Función 3					
...					

Para cada celda, estimar los requisitos de recursos, poner fechas a las tareas y determinar los productos a obtener

Actividades de gestión



- Calidad del producto
- Evaluación del riesgo
- Métricas
- Estimación de costes
- Confección de agendas
- Comunicación cliente
- Personal
- Otros recursos
- Monitorización del proyecto

Tema 2. Estimación de costes

-  Introducción
-  Productividad
-  Técnicas de estimación
-  Modelo algorítmico de costes
-  Duración y personal del proyecto

Bibliografía

Captítulo 26. Software cost estimation. Software
Ingeniering Sommerville 7^a edición.

Capítulo 18. Métricas del software. Ingeniería del
software. 4^a edición. Roger S. Pressman.

Introducción



¿Qué es la estimación de costes?

Consiste en predecir los **recursos** (monetarios, temporales, humanos, materiales, ...) necesarios para llevar a cabo el proceso de desarrollo del software.



Cuestiones fundamentales:

- ¿Cuánto esfuerzo es necesario para completar una actividad?
- ¿Cuánto tiempo se necesita para completar una actividad?
- ¿Cuál es el coste total de una actividad?

Componentes de coste

- Costes hardware y software
- Costes de viajes y aprendizaje
- Costes de **esfuerzo** (factor dominante casi siempre)
 - sueldo ingenieros del proyecto
 - gastos seguros y seguridad social
- Otros costes:
 - costes de alquiler, calefacción y luz
 - costes de redes y comunicaciones
 - costes de recursos compartidos (p.e. librería, personal del restaurante, etc.)

Factores de coste

Factor	Description
Market opportunity	A development organisation may quote a low price because it wishes to move into a new segment of the software market. Accepting a low profit on one project may give the opportunity of more profit later. The experience gained may allow new products to be developed.
Cost estimate uncertainty	If an organisation is unsure of its cost estimate, it may increase its price by some contingency over and above its normal profit.
Contractual terms	A customer may be willing to allow the developer to retain ownership of the source code and reuse it in other projects. The price charged may then be less than if the software source code is handed over to the customer.
Requirements volatility	If the requirements are likely to change, an organisation may lower its price to win a contract. After the contract is awarded, high prices may be charged for changes to the requirements.
Financial health	Developers in financial difficulty may lower their price to gain a contract. It is better to make a small profit or break even than to go out of business.

Productividad (I)

- La productividad de un programador es una medida de la "velocidad" a la que los ingenieros implicados en el desarrollo del software producen dicho **software** y su **documentación** asociada
- Es necesario estimar la productividad:
 - para realizar las estimaciones necesarias en el proyecto
 - para evaluar si un proceso o mejoras en la tecnología son efectivas.

Productividad (II)

$$\text{Productividad} = \frac{\text{atributos del software}}{\text{esfuerzo total de desarrollo}}$$

Tipos de medidas:

- Relacionadas con el tamaño (líneas de código)
- Relacionadas con la funcionalidad (puntos de función, puntos de objeto)

Líneas de código



¿Qué es una línea de código?

- Es una medida propuesta inicialmente cuando los programas se escribían en tarjetas, con una línea por tarjeta
- Actualmente los lenguajes permiten escribir varias sentencias en una línea, o una misma sentencia en varias líneas
- Se debe decidir qué programas deberían contarse como parte del sistema
- Asumen una relación lineal entre el tamaño y el volumen de documentación

Comparaciones de productividad

- Cuanto mayor sea la expresividad del lenguaje, más baja será su productividad aparente.
- Cuánto más líneas de código emplee el programador, mayor será su productividad

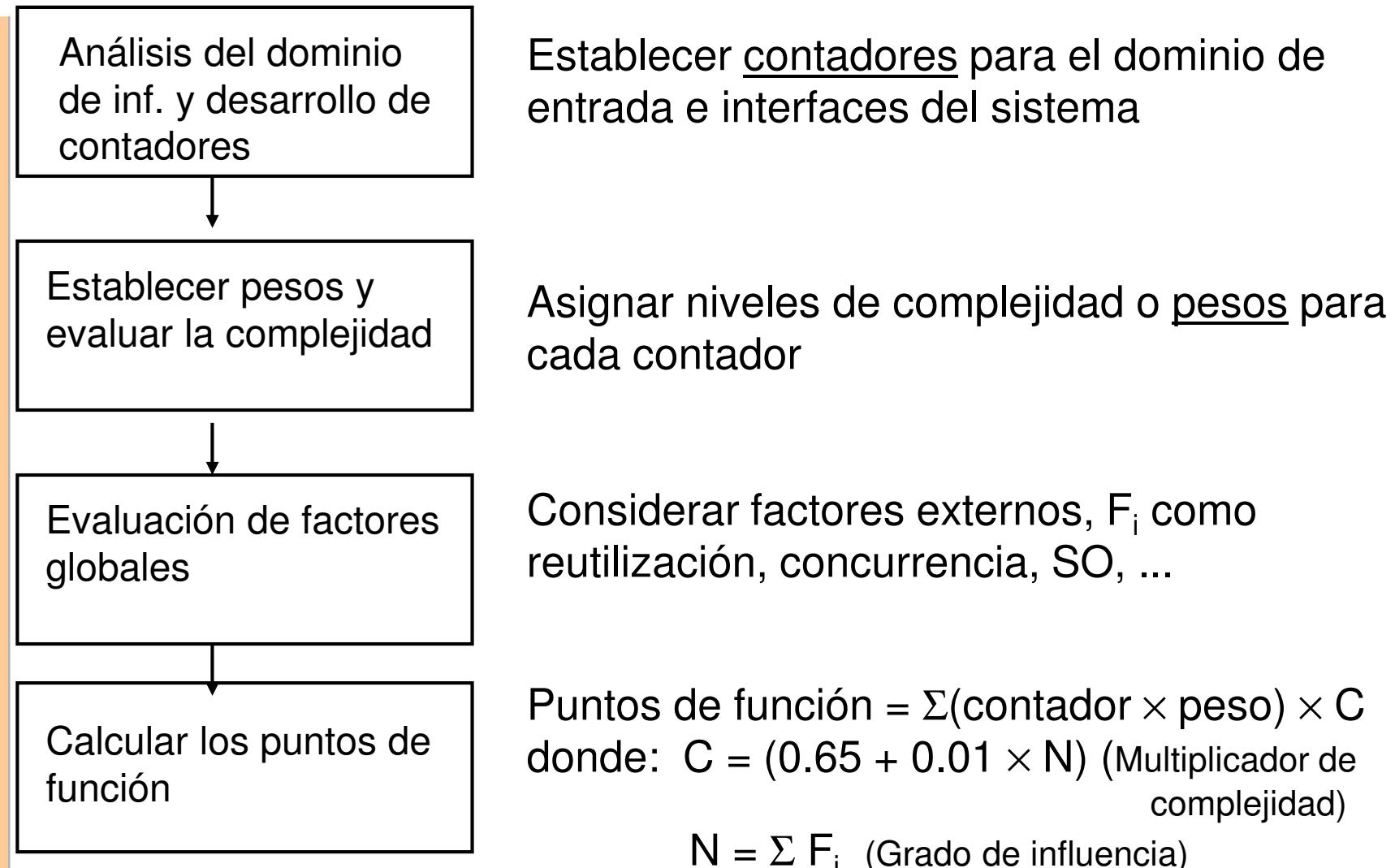
	Analysis	Design	Coding	Testing	Documentation
Assembly code	3 weeks	5 weeks	8 weeks	10 weeks	2 weeks
High-level language	3 weeks	5 weeks	8 weeks	6 weeks	2 weeks
	Size		Effort	Productivity	
Assembly code	5000 lines		28 weeks	714 lines/month	
High-level language	1500 lines		20 weeks	300 lines/month	

Comparar la productividad utilizando lenguajes diferentes de programación puede llevar a conclusiones erróneas respecto a la productividad de los programadores

Puntos de función

-  Basados en una combinación de características del programa
 - entradas y salidas externas
 - interacciones de usuario
 - interfaces externas
 - ficheros usados por el sistema
-  Se asocia un peso con cada uno de ellos
-  Los puntos de función se calculan multiplicando cada factor por su peso y sumando todos ellos

Cálculo de los puntos de función(I)



Cálculo de los puntos de función(II)

<u>Parámetro de medida</u>	<u>Contador</u>	<u>Factor de peso</u>			
		Simp.	Med.	Compl.	
Nº entradas usuario	<input type="text"/>	× 3	4	6	= <input type="text"/>
Nº salidas usuario	<input type="text"/>	× 4	5	7	= <input type="text"/>
Nº peticiones usuario	<input type="text"/>	× 3	4	6	= <input type="text"/>
Nº ficheros	<input type="text"/>	× 3	4	6	= <input type="text"/>
Nº interfaces externas	<input type="text"/>	× 7	10	15	= <input type="text"/>
Total contadores	—————				<input type="text"/>
Multiplicador de complejidad	—————				<input type="text"/>
Puntos de función	—————				<input type="text"/>

Ventajas de los puntos de función

- Son independientes del lenguaje de programación
- Pueden calcularse a partir de la especificación
- Usa información del dominio del problema
- Resulta más fácil a la hora de reusar componentes
- Se encamina a aproximaciones orientadas a objetos

Puntos de función y estimación

- Los puntos de función (FP) pueden usarse para estimar el **número de líneas de código** (LOC) dependiendo del número medio de LCDs por PF para un lenguaje dado (AVC)
 - LOC = AVC * número de puntos de función
 - AVC es un factor dependiente del lenguaje que varía desde 200-300 para lenguaje ensamblador hasta 2-40 para un lenguaje 4GL
- Los puntos de función son muy **subjetivos**.
Dependen del estimador.

Puntos de objeto

-  Los puntos de objeto son una medida alternativa relacionada con la funcionalidad cuando se utilizan lenguajes 4GLs o similares para el desarrollo
-  Los puntos de objeto **NO** son clases de objetos
-  El número de puntos de objeto en un programa es una estimación ponderada de:
 - El número de pantallas que son visualizadas por separado
 - El número de informes que se producen por el sistema
 - El número de módulos 3GL que deben desarrollarse para complementar el código 4GL

Estimación de puntos de objeto

- Son más fáciles de estimar a partir de una especificación que los puntos de función, ya que solamente consideran pantallas, informes y módulos 3GL
- Por lo tanto pueden estimarse en fases tempranas del proceso de desarrollo. En estas etapas resulta muy difícil estimar el número de líneas de código de un sistema

Factores que afectan la product.

-  Experiencia en el dominio de la aplicación
-  Calidad del proceso
-  Tamaño del proyecto
-  Tecnología de soporte
-  Entorno de trabajo

Calidad y productividad

- Todas las métricas basadas en volumen/unidad de tiempo son engañosas debido a que no tienen en cuenta la calidad
- La productividad puede incrementarse generalmente a costa de la calidad
- No está claro cómo la productividad y las métricas de calidad están relacionadas
- Las métricas de productividad deberían usarse únicamente como guía

Técnicas de estimación (I)

 No existe una forma simple de obtener estimaciones exactas del esfuerzo requerido para desarrollar un sistema software

- Las estimaciones iniciales se basan en información incompleta en la definición de requerim. del usuario
- El software puede tener que ejecutarse sobre ordenadores no usuales o usar nuevas tecnologías
- Puede desconocerse a la gente que interviene en el proy.

Técnicas de estimación (II)

 Modelado algorítmico de costes

 Juicio experto

 Estimación por analogía

 Ley de Parkinson

 *Pricing to win*

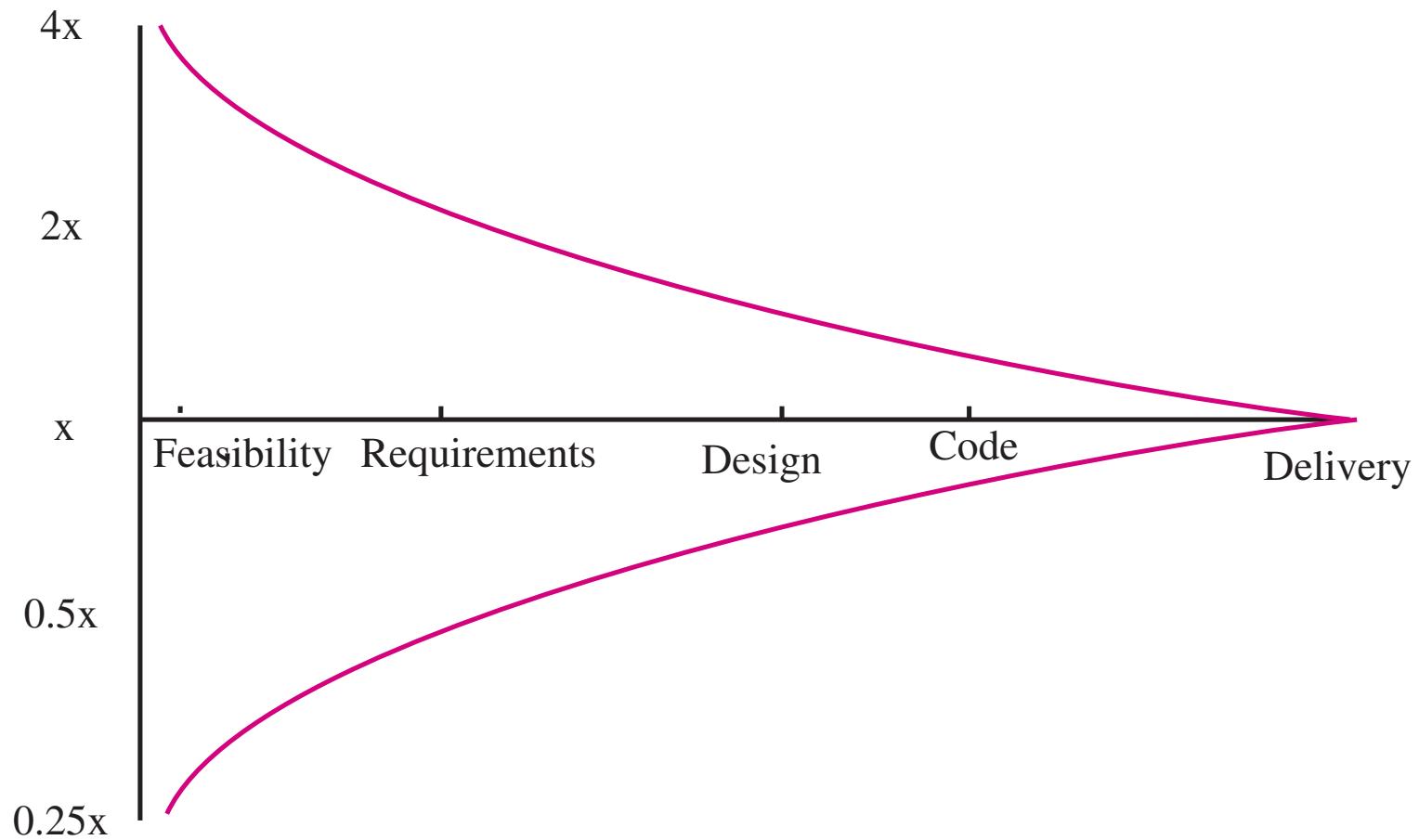
Modelado algorítmico de costes

- Es una aproximación que utiliza fórmulas obtenidas a partir de información histórica. Suele basarse en el tamaño del software
- La mayoría de modelos tienen una componente exponencial (los costes no crecen normalmente de forma lineal con el tamaño del proyecto)
 - $\text{Esfuerzo} = A \times \text{Tamaño}^B \times M$

Exactitud de la estimación

- El tamaño de un sistema software puede conocerse con exactitud solamente cuando está terminado
- Algunos factores que influyen en el tamaño final son:
 - Uso de COTs y componentes
 - Lenguaje de programación utilizado
 - Distribución del sistema
- La estimación del tamaño se realiza de forma más exacta a medida que el desarrollo del sistema progresá

Estimar la incertidumbre



Juicio experto

-  Uno o más expertos, tanto en desarrollo de software como en el dominio de la aplicación usan su experiencia para predecir los costes de software. Se realizan iteraciones hasta que se alcanza un consenso
-  **Ventajas:** Método de estimación relativamente barato. Puede ser bastante exacto si los expertos tienen experiencia directa en sistemas similares
-  **Desventajas:** ¡Muy impreciso si no se dispone de los expertos adecuados!

Estimación por analogía

- El coste de un proyecto se calcula por comparación con proyectos similares en el mismo dominio de aplicación
- **Ventajas:** Bastante preciso si se disponen de datos de proyectos previos
- **Inconvenientes:** Imposible de realizar sin no se han abordado proyectos comparables. Necesita un mantenimiento sistemático de una base de datos

Ley de Parkinson

-  Los costes del proyecto están en función de los recursos disponibles, utilizando todo el tiempo permitido.
-  **Ventajas:** No realiza presupuestos "abultados"
-  **Inconvenientes:** El sistema normalmente no termina

Pricing to win

-  El coste del proyecto está en función de lo que el cliente está dispuesto a pagar
-  **Ventajas:** La empresa desarrolladora consigue el contrato
-  **Inconvenientes:** La probabilidad de que el cliente obtenga el sistema que quiere es pequeña. Los costes no reflejan realmente el trabajo requerido

Estimación ascend. y descend.

- Cualquiera de estas aproximaciones puede utilizarse de forma ascendente o descendente
 - Descendente
 - Comienza a nivel de sistema y evalúa la totalidad de funcionalidades y cómo éstas se subdividen en subsistemas
 - Ascendente
 - Comienza a nivel de componentes y estima el esfuerzo requerido para cada componente. Dichos esfuerzos se añaden a la estimación final

Estimación descendente

- Se puede usar sin conocer la arquitectura ni los componentes que formarán parte del sistema
- Tiene en cuenta costes tales como integración, gestión de configuraciones y documentación
- Puede infra-estimar costes relacionados con la resolución de problemas técnicos de bajo nivel difíciles de resolver

Estimación ascendente

- Se puede usar cuando la arquitectura del sistema es conocida y los componentes han sido identificados
- Proporciona estimaciones bastante exactas si el sistema se ha diseñado con detalle
- Puede infra-estimar costes a nivel de sistema, tales como integración y documentación

Comparando métodos

- Cada método tiene sus ventajas e inconvenientes
- La estimación debería basarse en varios métodos
- Si el resultado de aplicar varios de ellos difiere mucho, es que no se dispone de suficiente información
- Muchas veces el método *Pricing to win* es el único aplicable

Modelo COCOMO

Constructive **C**Ost **M**odel

-  Es un modelo empírico basado en la experiencia con proyectos (grandes)
-  Es un método bien documentado, cuya primera versión se publicó en 1981
-  La última versión, COCOMO 2, tiene en cuenta diferentes aproximaciones de desarrollo, reutilización, etc.

COCOMO 81

Project complexity	Formula	Description
Simple	$PM = 2.4 (\text{KDSI})^{1.05} \times M$	Well-understood applications developed by small teams.
Moderate	$PM = 3.0 (\text{KDSI})^{1.12} \times M$	More complex projects where team members may have limited experience of related systems.
Embedded	$PM = 3.6 (\text{KDSI})^{1.20} \times M$	Complex projects where the software is part of a strongly coupled complex of hardware, software, regulations and operational procedures.

Niveles COCOMO 2

- COCOMO 2 es un modelo de tres niveles que permite estimaciones cada vez más detalladas y que pueden realizarse a la vez que progresá el desarrollo del proyecto
 - Nivel inicial de prototipado
 - Estimaciones realizadas con puntos de objeto y una fórmula simple para el cálculo del esfuerzo
 - Nivel inicial de diseño
 - Estimaciones realizadas con puntos de función convertidas en líneas de código
 - Nivel post-arquitectura
 - Estimaciones basadas en líneas de código fuente

Nivel inicial prototipado

- Soporta proyectos con prototipado y proyectos que hacen uso intensivo de la reutilización
- Basado en estimaciones estándar de la productividad del desarrollador en puntos-objeto/mes
- Tiene en cuenta el uso de herramientas CASE
- La fórmula es:
 - $OPM = (NOP \times (1 - \%reuse/100)) / PROD$
 - OPM es el esfuerzo en personas-mes, NOP es el número de puntos de objeto, y PROD es la productividad

Productiv. de puntos de objeto



La productividad (PROD) depende de:

- La experiencia y capacidad del desarrollador
- Las capacidades de la herramienta CASE utilizada

Developer's experience and capability	Very low	Low	Nominal	High	Very high
ICASE maturity and capability	Very low	Low	Nominal	High	Very high
PROD (NOP/month)	4	7	13	25	50

Nivel inicial de diseño

- Las estimaciones pueden hacerse después de que los requerimientos hayan sido establecidos
- Basado en las fórmulas estándar para métodos algorítmicos
 - $OPM = A \times \text{Tamaño}^B \times M + PM_m$ en donde
 - $O\text{M} = PERS \times RCPX \times RUSE \times PDIF \times PREX \times FCIL \times SCED$
 - $O\text{PM}_m = (\text{ASLOC} \times (\text{AT}/100)) / \text{ATPROD}$
 - $O\text{A} = 2.5$ según la calibración inicial, **Tamaño** se da en KLOC, **B** varía desde 1.1 hasta 1.24 dependiendo de la novedad del proyecto, la flexibilidad del desarrollo, la gestión de riesgos, y la madurez del proceso

Multiplicadores (**M**)

-  Los multiplicadores reflejan la capacidad de los desarrolladores, requerim. no funcionales, la familiaridad con la plataforma de desarrollo, etc.
 - RCPX - fiabilidad de producto y complejidad
 - RUSE - reutilización requerida
 - PDIF - dificultad de la plataforma
 - PREX - experiencia del personal
 - PERS - capacidad del personal
 - SCED - agenda requerida
 - FCIL - facilidades de soporte de grupo
-  **PM** refleja la cantidad de código generada automáticamente

Nivel post-arquitectura

Uso de la misma fórmula que la estimación inicial de diseño

$$\textcircled{O} \quad PM = A \times \text{Tamaño}^B \times M + PM_m$$

Se ajusta la estimación de **tamaño** para que tenga en cuenta

O La volatilidad de los requerimientos

O Grado de posible reutilización

$$\textcircled{O} \quad \text{ESLOC} = \text{ASLOC} \times (\text{AA} + \text{SU} + 0.4\text{DM} + 0.3\text{CM} + 0.3\text{IM})/100$$

ESLOC es el número de líneas de código nuevo. **ASLOC** es el número de líneas de código reusable que debe modificarse, **DM** es el porcentaje de diseño modificado, **CM** es el porcentaje de código que se modifica, **IM** es el porcentaje del esfuerzo original de integración del software reusado.

SU es un factor basado en la interpretación del coste del software, **AA** es un factor que refleja los costes de evaluación iniciales para decidir si el software puede reutilizarse.

El término exponente (**B**)

 Depende de 5 factores de escala. La suma de dichos factores se divide por 100 y se añade a 1.01

 Ejemplo

-  Antecedentes - proyecto nuevo - 4
-  Flexibilidad desarrollo - no implicación cliente - Muy alto - 1
-  Arquitectura/resolución riesgos - No análisis de riesgos - Muy bajo - 5
-  Cohesión del grupo - nuevo grupo - nominal - 3
-  Madurez proceso - algún control - nominal - 3

 El factor de escala es 1.17

Factores de escala de exponente

Scale factor	Explanation	(B)
Precedentedness	Reflects the previous experience of the organisation with this type of project. Very low means no previous experience, Extra high means that the organisation is completely familiar with this application domain.	
Development flexibility	Reflects the degree of flexibility in the development process. Very low means a prescribed process is used; Extra high means that the client only sets general goals.	
Architecture/risk resolution	Reflects the extent of risk analysis carried out. Very low means little analysis, Extra high means a complete a thorough risk analysis.	
Team cohesion	Reflects how well the development team know each other and work together. Very low means very difficult interactions, Extra high means an integrated and effective team with no communication problems.	
Process maturity	Reflects the process maturity of the organisation. The computation of this value depends on the CMM Maturity Questionnaire but an estimate can be achieved by subtracting the CMM process maturity level from 5.	

Multiplicadores (**M**)

 Atributos del producto

 Atributos del ordenador

 Atributos del personal

 Atributos del proyecto

Conductores de coste del proy.

Product attributes			
RELY	Required system reliability	DATA	Size of database used
CPLX	Complexity of system modules	RUSE	Required percentage of reusable components
DOCU	Extent of documentation required		
Computer attributes			
TIME	Execution time constraints	STOR	Memory constraints
PVOL	Volatility of development platform		
Personnel attributes			
ACAP	Capability of project analysts	PCAP	Programmer capability
PCON	Personnel continuity	AEXP	Analyst experience in project domain
PEXP	Programmer experience in project domain	LTEX	Language and tool experience
Project attributes			
TOOL	Use of software tools	SITE	Extent of multi-site working and quality of site communications
SCED	Development schedule compression		

Efectos de los conduct. de coste

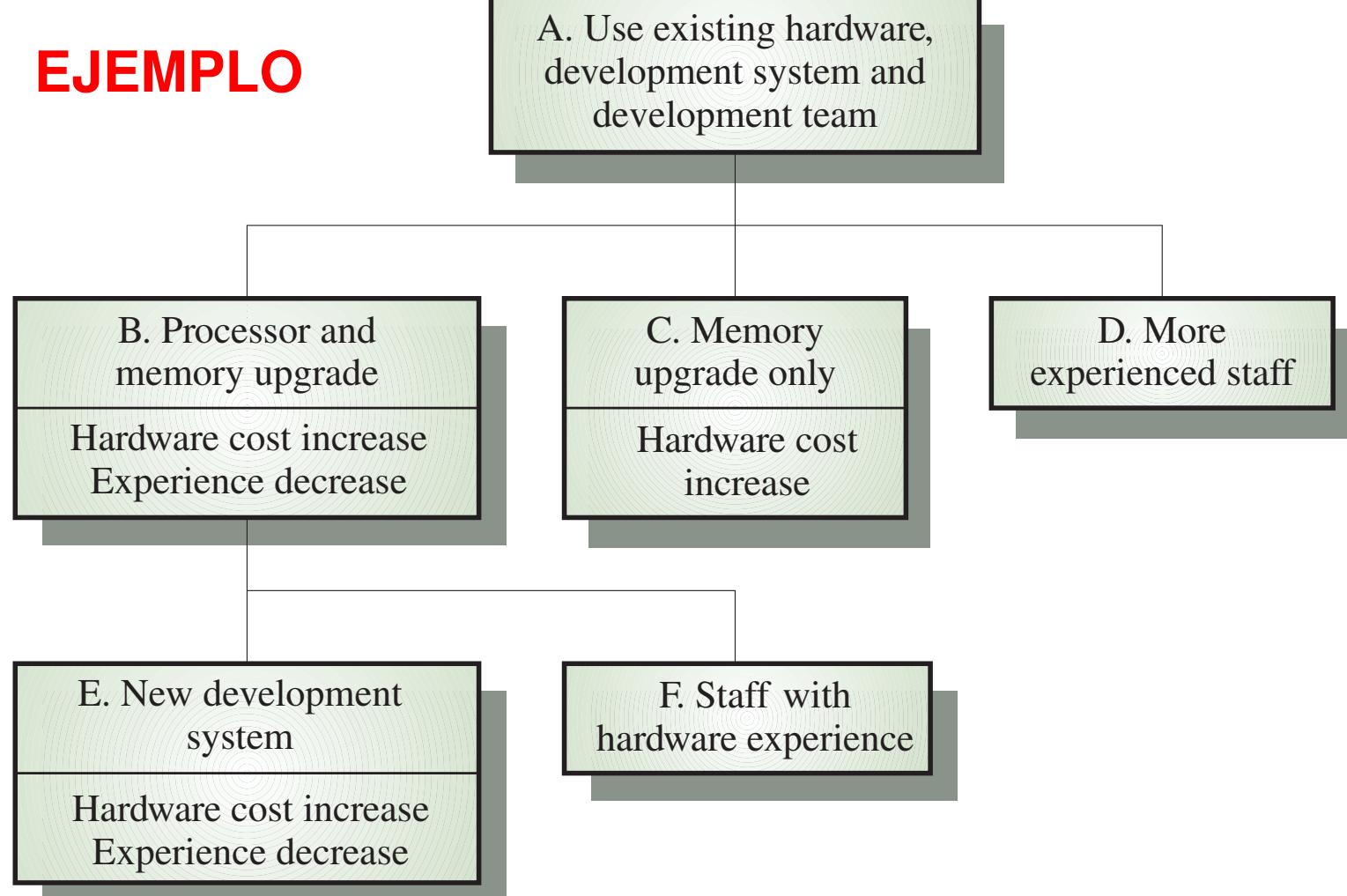
Exponent value	1.17	(M)
System size (including factors for reuse and requirements volatility)	128, 000 DSI	
Initial COCOMO estimate without cost drivers	730 person-months	
Reliability	Very high, multiplier = 1.39	
Complexity	Very high, multiplier = 1.3	
Memory constraint	High, multiplier = 1.21	
Tool use	Low, multiplier = 1.12	
Schedule	Accelerated, multiplier = 1.29	
Adjusted COCOMO estimate	2306 person-months	
Reliability	Very low, multiplier = 0.75	
Complexity	Very low, multiplier = 0.75	
Memory constraint	None, multiplier = 1	
Tool use	Very high, multiplier = 0.72	
Schedule	Normal, multiplier = 1	
Adjusted COCOMO estimate	295 person-months	

Planificación del proyecto

- Los modelos algorítmicos de costes proporcionan una base para la planificación del proyecto en tanto que permiten comparar estrategias alternativas
- Ejemplo: desarrollo de un sistema espacial empotrado
 - Debe ser fiable
 - Debe tener un peso mínimo (número de chips)
 - Multiplicadores de fiabilidad y restricciones del ordenador > 1
- Componentes de coste
 - Hardware destino
 - Plataforma de desarrollo
 - Esfuerzo requerido

Opciones de gestión

EJEMPLO



Gestión de opciones de coste

EJEMPLO

Option	RELY	STOR	TIME	TOOLS	LTEX	Total effort	Software cost	Hardware cost	Total cost
A	1.39	1.06	1.11	0.86	1	63	949393	100000	1049393
B	1.39	1	1	1.12	1.22	88	1313550	120000	1402025
C	1.39	1	1.11	0.86	1	60	895653	105000	1000653
D	1.39	1.06	1.11	0.86	0.84	51	769008	100000	897490
E	1.39	1	1	0.72	1.22	56	844425	220000	1044159
F	1.39	1	1	1.12	0.84	57	851180	120000	1002706

Selección de opciones

 Opción D (usa más personal con experiencia) parece la mejor alternativa

○ Sin embargo tiene un alto riesgo asociado ya que personal con experiencia puede ser difícil de encontrar

 Opción C (actualización memoria) tiene un menor ahorro de costes, pero un riesgo muy bajo

En conjunto, el modelo revela la importancia del personal con experiencia en el desarrollo del software

Duración y personal del proyecto

- Además de la estimación del esfuerzo, se debe estimar el tiempo requerido para terminar el proyecto, así como el personal necesario
- La duración del proyecto puede estimarse mediante la fórmula de COCOMO 2
 - $TDEV = 3 \times (PM)^{(0.33+0.2*(B-1.01))}$
 - PM es el esfuerzo.

La duración es independiente del número de gente que trabaje en el proyecto (depende del esfuerzo total invertido en el proyecto)

Puntos clave

-  Es necesario estimar costes: (esfuerzo, tiempo de desarrollo y número de recursos)
-  La productividad es un factor a tener en cuenta a la hora de realizar estimaciones
-  Existen varias técnicas de estimación de costes. La estimación algorítmica de costes es difícil al necesitar una estimación previa de atributos del producto terminado
-  Los modelos de estimación algorítmicos suponen una opción de análisis cuantitativo
-  El tiempo necesario para completar un proyecto no es proporcional al número de personas que trabajan en el mismo

Tema 3. Planificación de proyectos



- Planificación de proyectos
- Confección de agendas (*scheduling*)
- Gestión de riesgos

Bibliografía

- Captítulo 5. Project Management. Software Engineering Sommerville 7^a edición.
- Capítulo 5. Planificación de Proyectos software. Ingeniería del software. 4^a edición. Roger S. Pressman.
- Capítulo 6. Gestión del riesgo. Ingeniería del software. 4^a edición. Roger S. Pressman.
- Capítulo 7. Planificación temporal y seguimiento del proyecto. Ingeniería del software. 4^a edición. Roger S. Pressman.

Importancia de la gestión

- Se trata probablemente de la actividad de gestión que más tiempo consume
- Es una actividad que se realiza de forma **CONTINUADA** desde el concepto inicial del sistema, hasta su entrega. Los planes se deben revisar continuamente según esté disponible nueva información

Tipos de planes de proyectos

- Plan de calidad
- Plan de validación
- Plan de gestión de configuraciones
- Plan de mantenimiento
- Plan de gestión de personal

Proceso de planificación

Planificar es decidir de antemano:

- **QUÉ** hay que hacer
- **CÓMO** hay que hacerlo
- **CUÁNDO** se va a hacer
- **QUIÉN** lo va a hacer

ALGORITMO:

Establecer **restricciones** proy.

Hacer **evaluaciones** iniciales

Definir **hitos** y **entregas**

Mientras proy no acaba hacer

Confeccionar **agenda**

Iniciar trabajos agenda

Esperar

Revisar el progreso

Revisar estimaciones

Actualizar agenda

Re-negociar restricciones

Solucionar posibles problemas

Fin mientras

Estructura del plan del proyecto

- Introducción
- Organización del proyecto
- Análisis de riesgos
- Requerimientos hardware y software
- Estructura de actividades del proyecto
- Agenda del proyecto
- Mecanismos de monitorización y control

Organización de las actividades

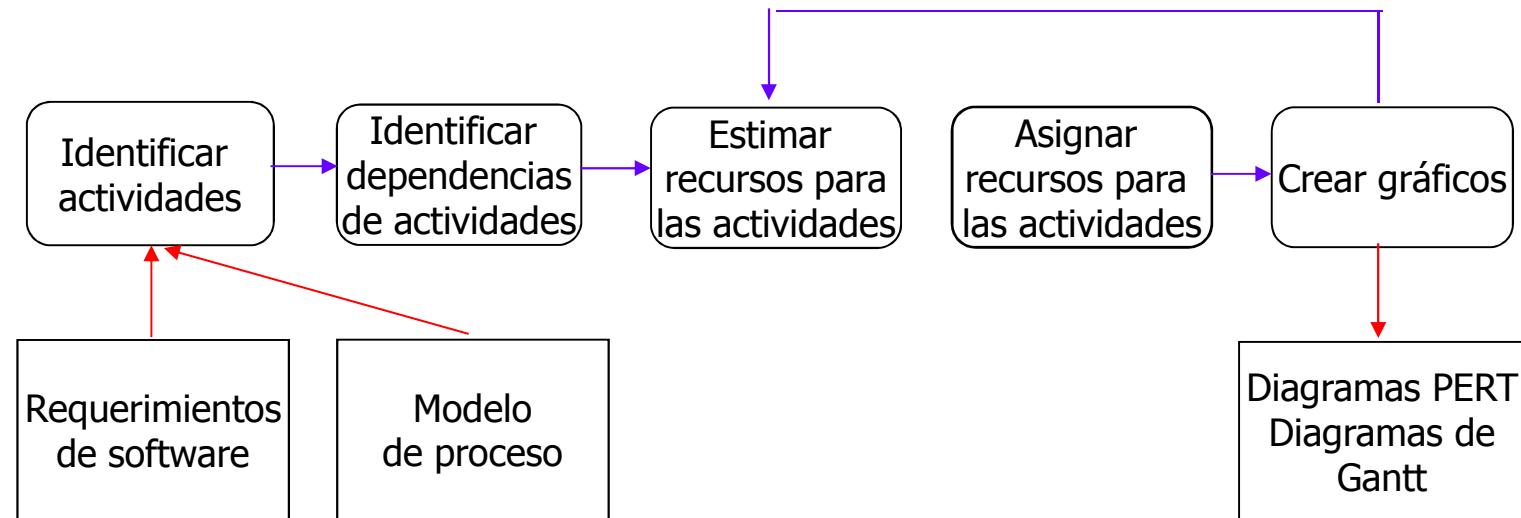
- Las actividades se deben organizar de forma que produzcan salidas "tangibles" válidas para la gestión del progreso del desarrollo
- Los HITOS (*milestones*) marcan el final de una actividad del proceso de desarrollo
- Las ENTREGAS (*deliverables*) son resultados del proyecto que se entregan a los clientes
- El proceso en cascada permite identificar de forma sencilla los **hitos** que marcan el progreso del proyecto

Scheduling

- Consiste en la organización temporal y asignación de recursos a las actividades de un proyecto.

- Pasos a seguir:
 - Determinación de las actividades a realizar
 - Asignación de tiempos estimados
 - Asignación de recursos
 - Organización temporal de las actividades

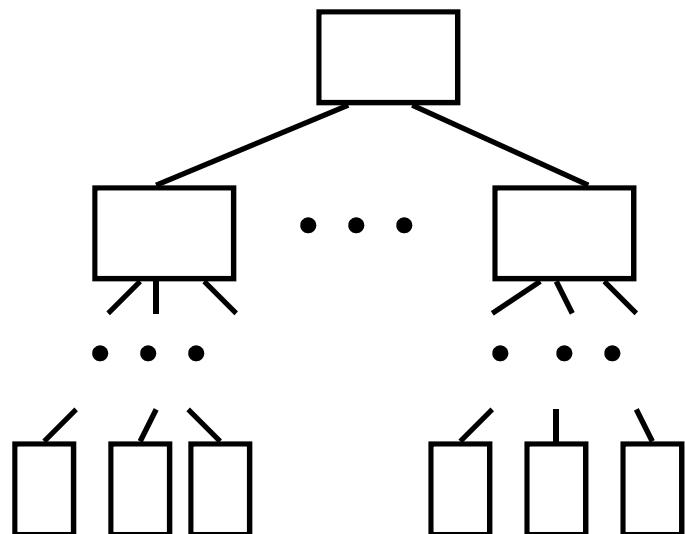
El proceso de *Scheduling*



Estructuración de las actividades

Agrupación de actividades por niveles.

Sirven para situar las actividades dentro de la organización del proyecto



00000 Nivel 0
10000 Subnivel 1
11000 Subnivel 1.1
11100 Actividad 1.1.1
11200 Actividad 1.1.2

20000 Subnivel 2

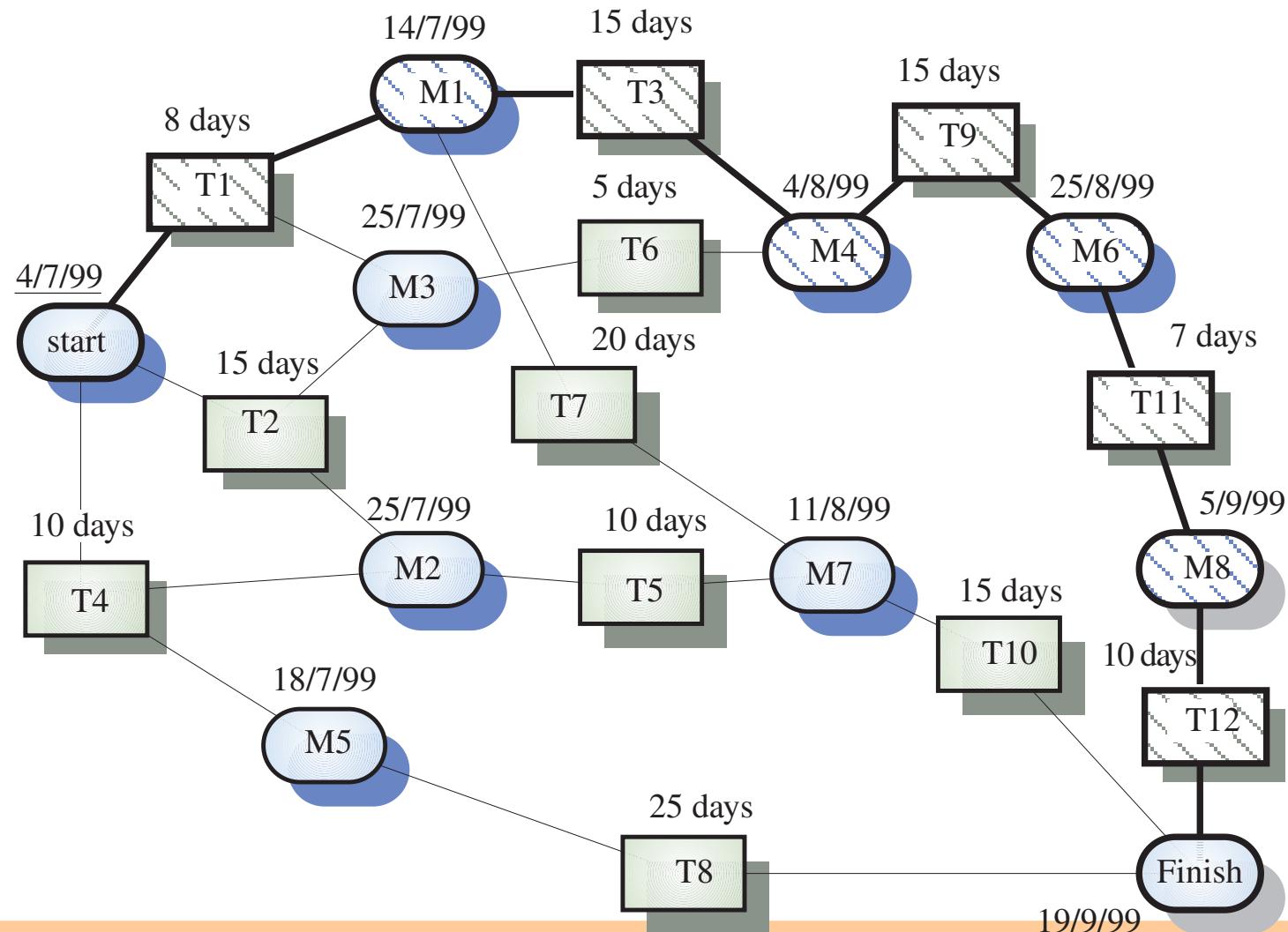
Representaciones gráficas

- Se utilizan para ilustrar la agenda del proyecto
- Permiten mostrar una vista de la división en tareas del proyecto. Las actividades no deberían ser demasiado "pequeñas" (alrededor de una semana o más)
- Los diagramas de actividades muestran las dependencias de las tareas y el camino crítico
- Los diagramas de barras muestran la agenda del proyecto

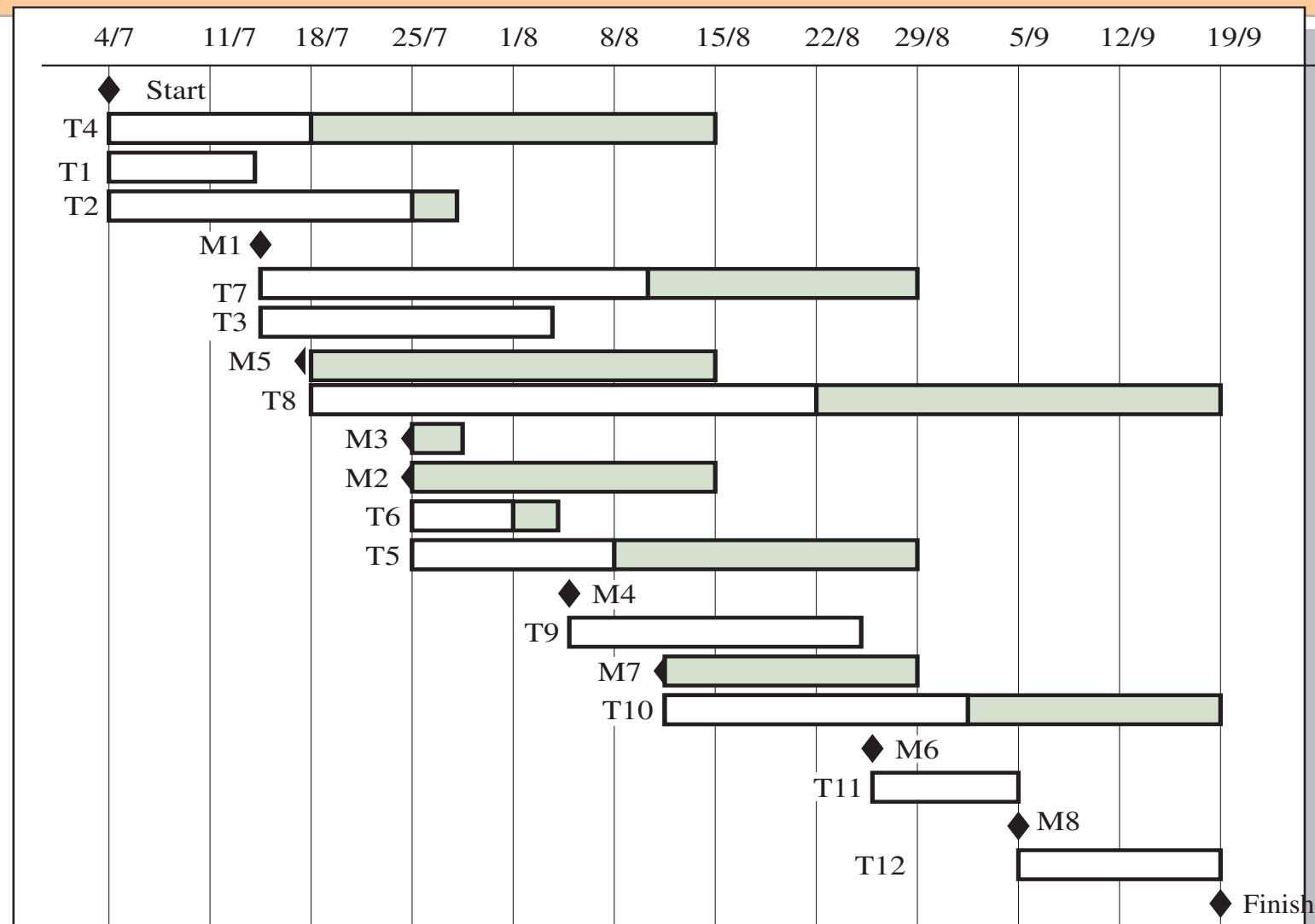
Duración de tareas y dependencias

Task	Duration (days)	Dependencies
T1	8	
T2	15	
T3	15	T1 (M1)
T4	10	
T5	10	T2, T4 (M2)
T6	5	T1, T2 (M3)
T7	20	T1 (M1)
T8	25	T4 (M5)
T9	15	T3, T6 (M4)
T10	15	T5, T7 (M7)
T11	7	T9 (M6)
T12	10	T11 (M8)

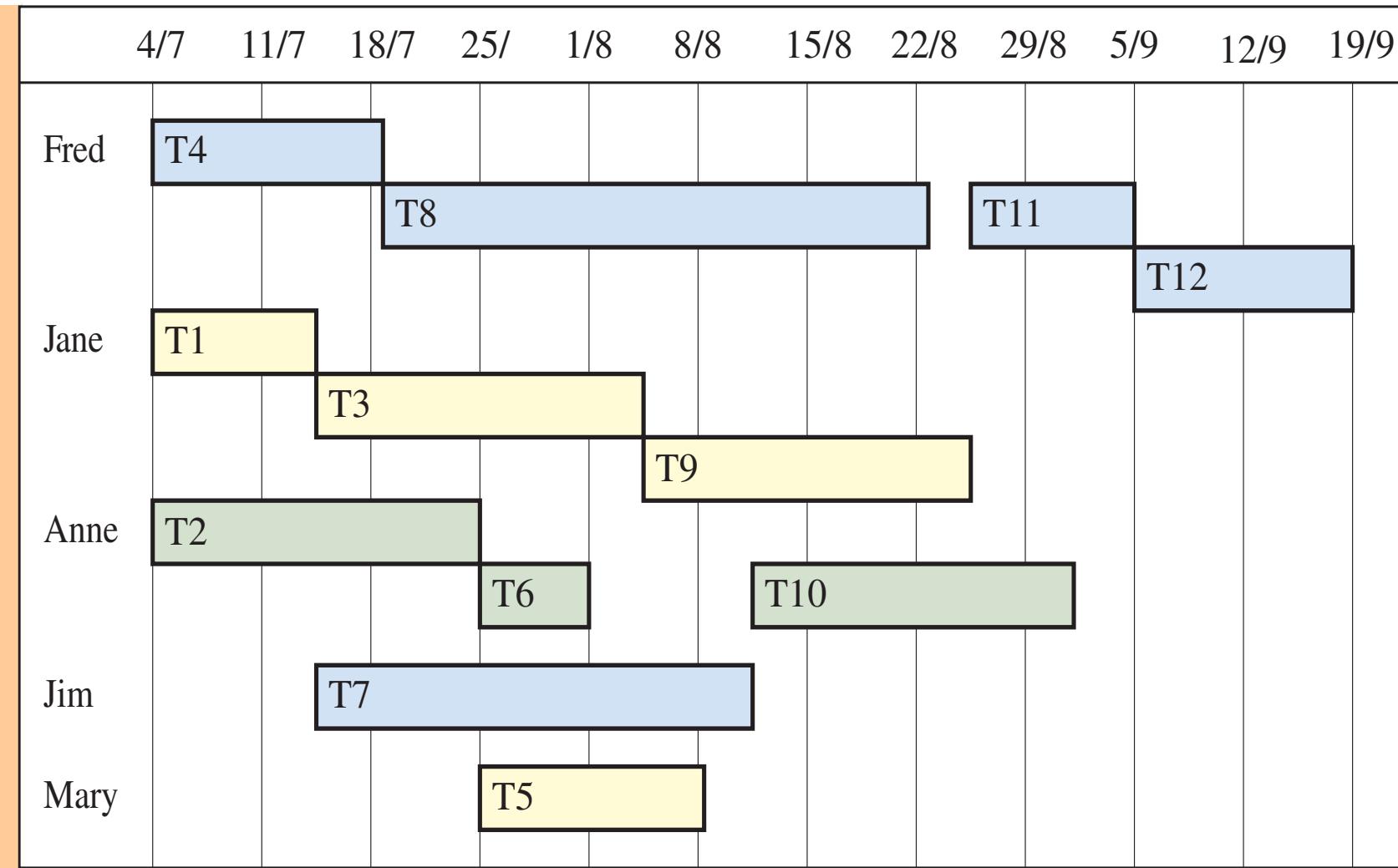
Red de actividades



Secuencia temporal actividades



Asignación de personal



Grafos PERT

Datos de entrada:

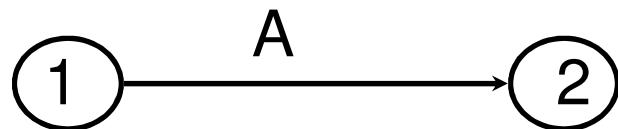
- Lista de precedencias del proyecto
- Asignación de tiempos y recursos a actividades

Proceso a realizar:

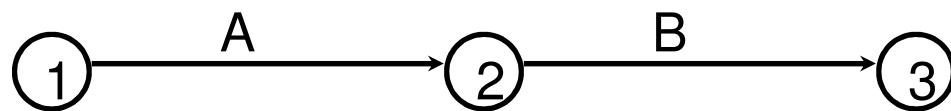
- Representar grafo
- Cálculo de tiempos “early” y “last”
- Cálculo de holguras
- Cálculo del camino crítico
- Confección de agenda

Precedencia entre actividades(I)

$G = (V, E)$, V = cto. de vértices \rightarrow sucesos
 E = cto. de aristas \rightarrow actividades

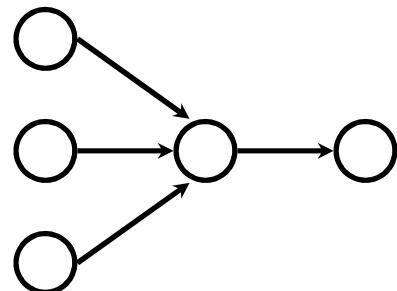


PRELACIÓN LINEAL

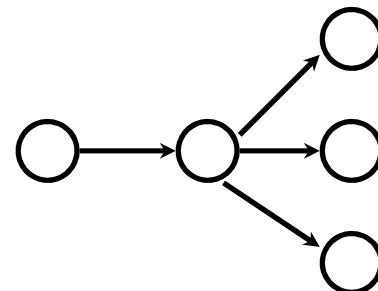


Precedencia entre actividades(II)

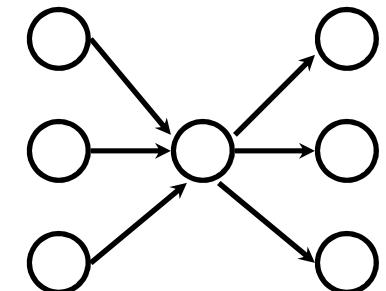
CONVERGENCIA



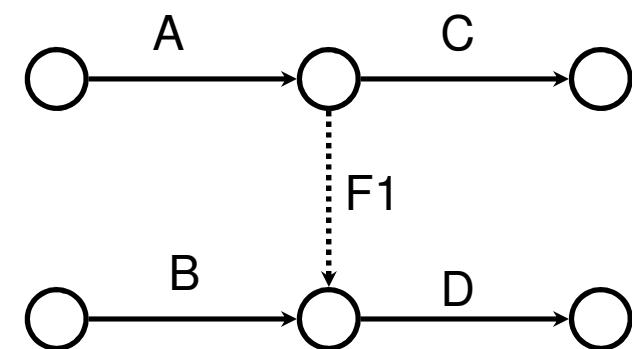
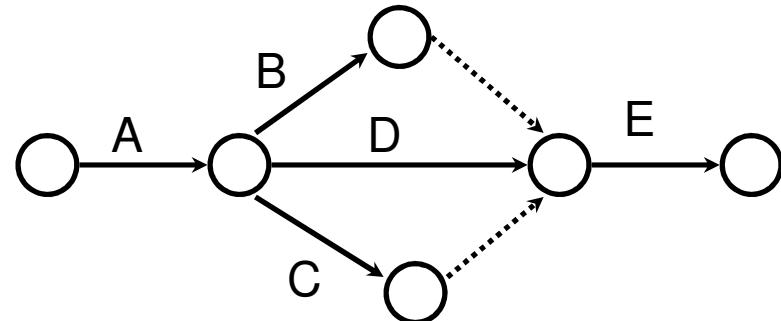
DIVERGENCIA



CONV-DIVER.



ACTIVIDADES FICTICIAS



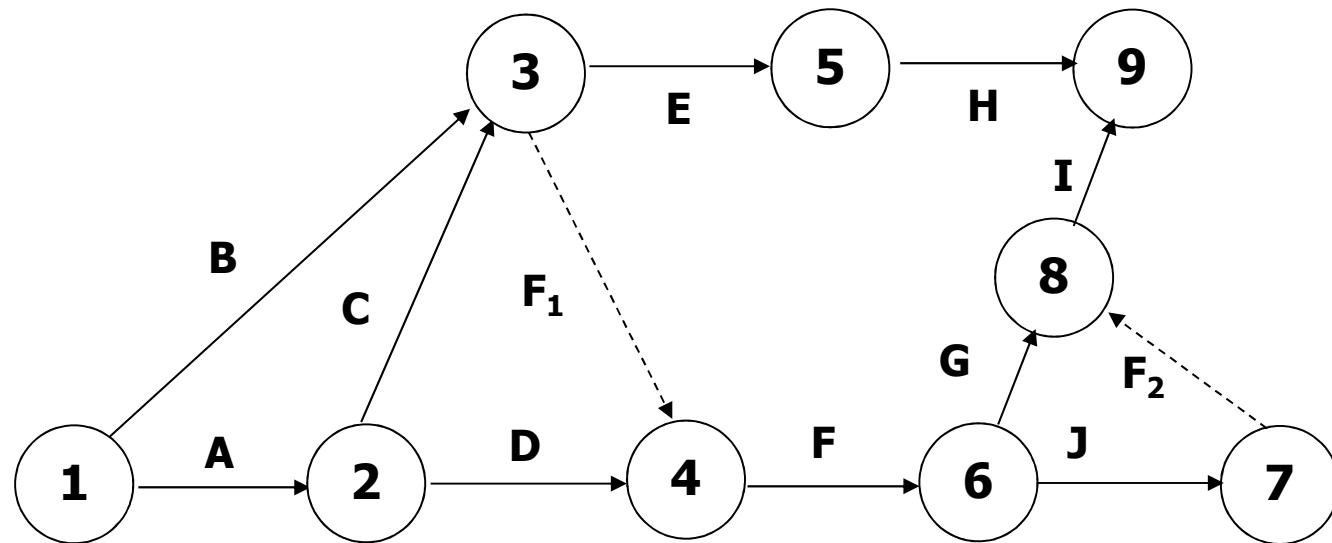
Precedencia entre actividades(III).

Ejercicio: Dibuja el grafo PERT.

Actividades	Precedentes	Duraciones
A	---	2
B	---	3
C	A	7
D	A	8
E	B, C	3
F	B, C, D	9
G	F	8
H	E	2
I	G, J	2
J	F	10

Precedencia entre actividades(VI).

Ejercicio: Dibuja el grafo PERT (SOLUCION)



Tiempos "early" y "last" (I)



Cálculo de tiempos más tempranos de ejecución

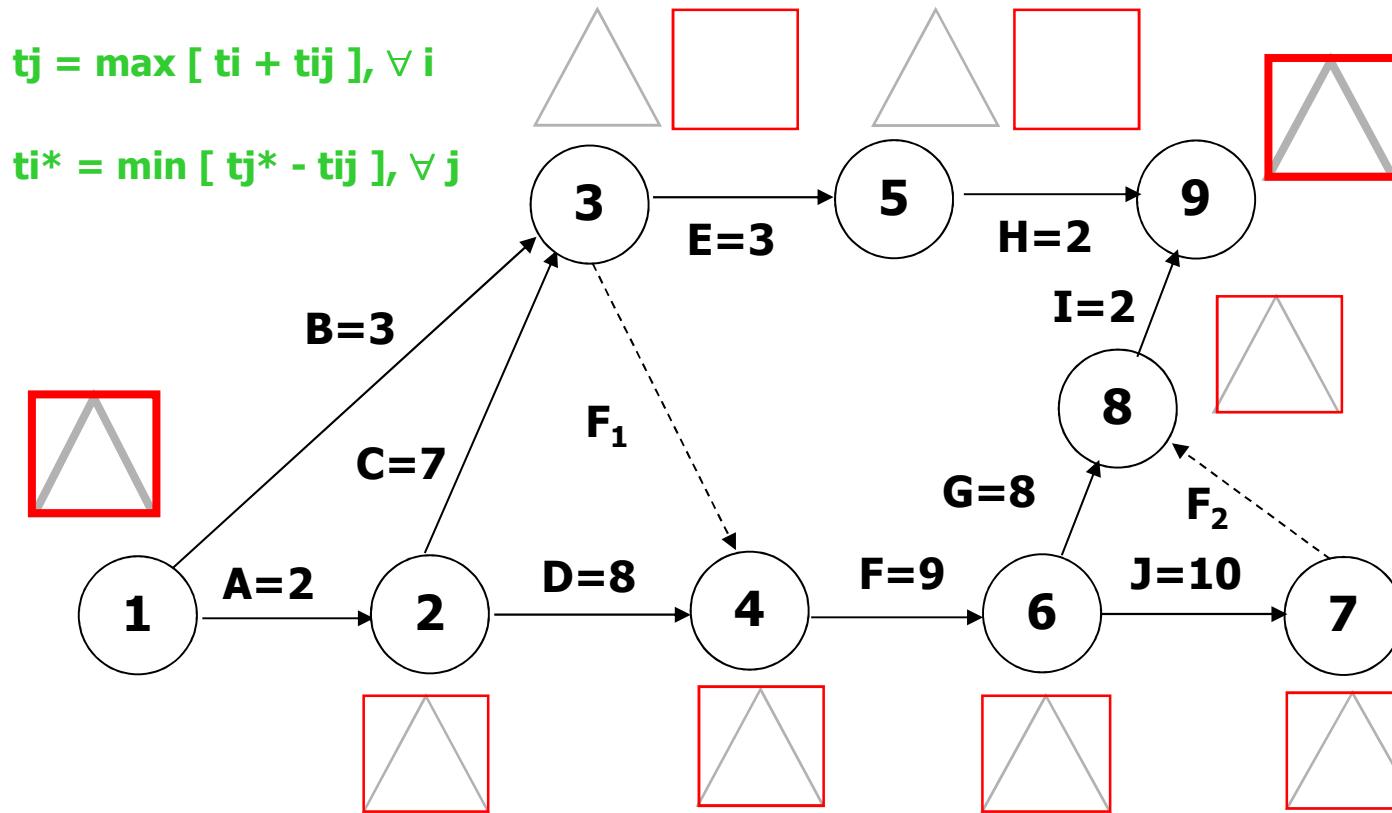
$$t_j = \max [t_i + t_{ij}] \text{ para todo } i \quad \triangle$$

Cálculo de tiempos más tardíos de ejecución

$$t_i^* = \min [t_j^* - t_{ij}] \text{ para todo } j \quad \square$$

Tiempos "early" y "last" (II).

Ejercicio: Calcula los tiempos last y early.

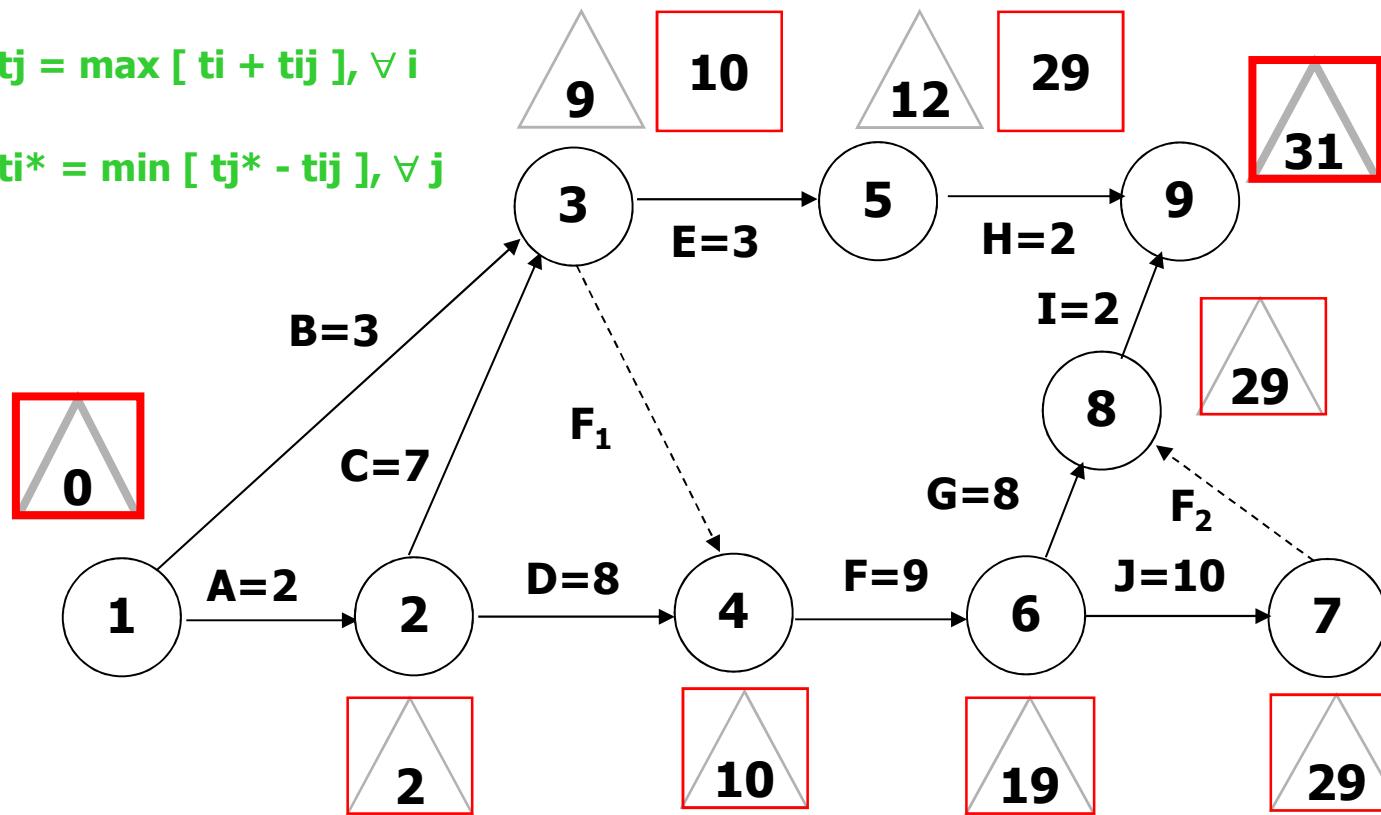


Tiempos "early" y "last" (II).

Ejercicio: Calcula los tiempos last y early (SOLUCION).

$$t_j = \max [t_i + t_{ij}], \forall i$$

$$t_{i^*} = \min [t_{j^*} - t_{ij}], \forall j$$



Holguras y camino crítico (I)

$$H_i = t_i^* - t_i$$

$$H_{ij}^T = t_j^* - t_i - t_{ij} \quad \text{Holgura total}$$

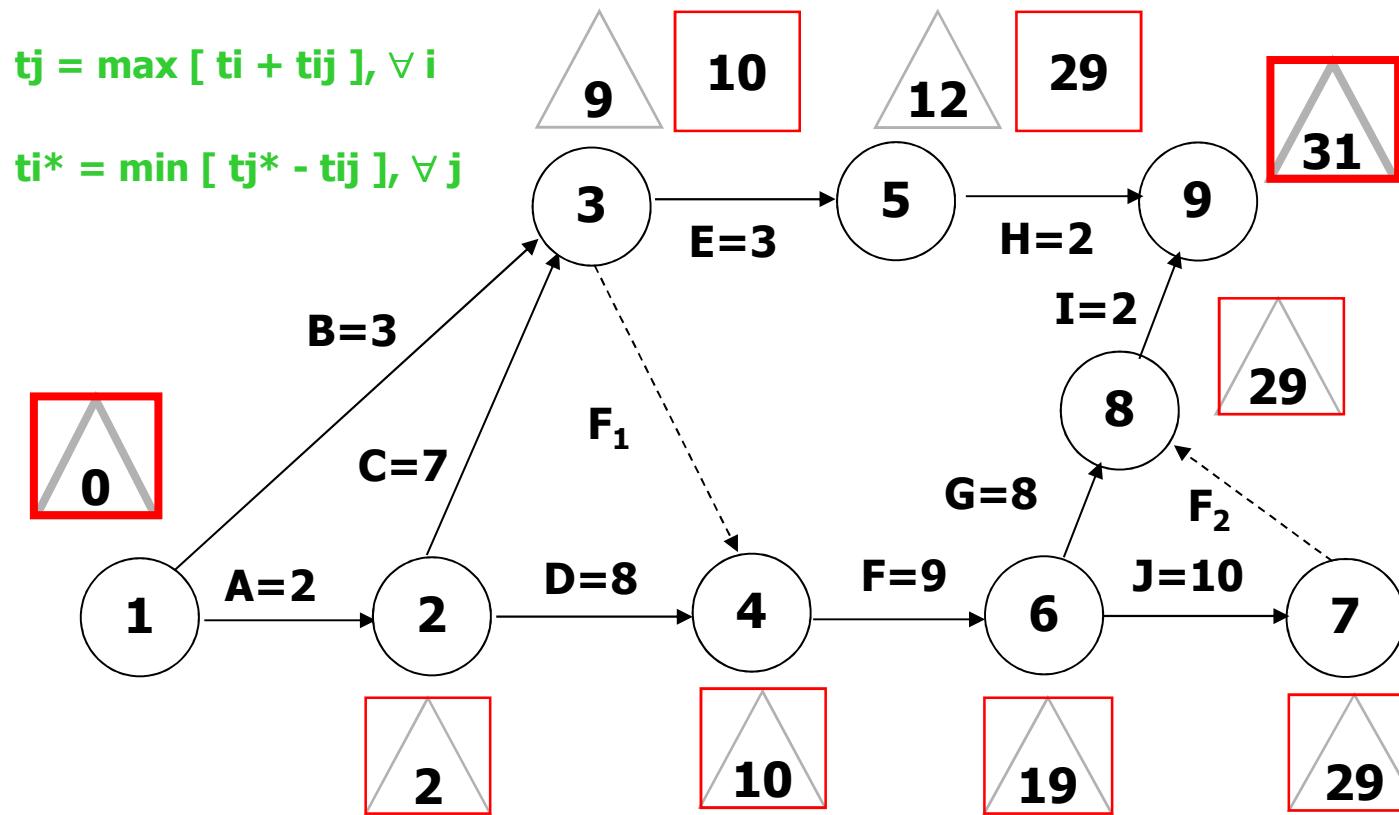
$$H_{ij}^L = t_j - t_i - t_{ij} \quad \text{Holgura libre}$$

Cálculo del camino crítico

- Camino más largo en el grafo
- Viene determinado por el **tiempo early** del suceso fin del proyecto
- Puede haber más de un camino crítico
- **TODAS** las actividades del camino crítico tienen holgura total = 0 (condición necesaria)

Holguras y camino crítico (II)

Ejercicio: Calcula las holguras y establece el camino crítico.

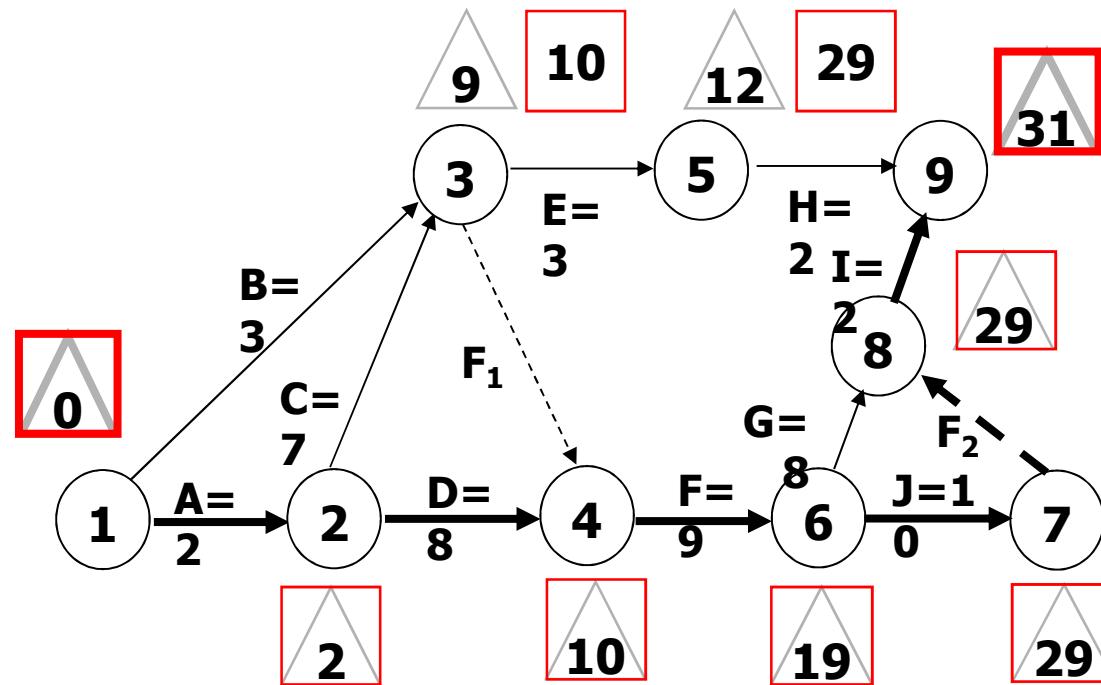


Holguras y camino crítico (III).

Ejercicio: Calcula las holguras y establece el camino crítico (SOLUCION).

$$H^T_{ij} = t_j^* - t_i - t_{ij}$$

ACTIVIDAD (i-j)	A (1-2)	B (1-3)	C (2-3)	D (2-4)	E (3-5)	F (4-6)	G (6-8)	H (5-9)	I (8-9)	J (6-7)
H^T_{ij}	0	7	1	0	17	0	2	17	0	0



Agenda (I)



Fecha de comienzo

$$\triangle_{ij} = t_i$$

Más temprana

$$\triangle_{ij}^* = t_j^* - t_{ij}$$

Más tardía



Fecha de fin

$$\nabla_{ij} = t_i + t_{ij}$$

Más temprana

$$\nabla_{ij}^* = t_j^*$$

Más tardía

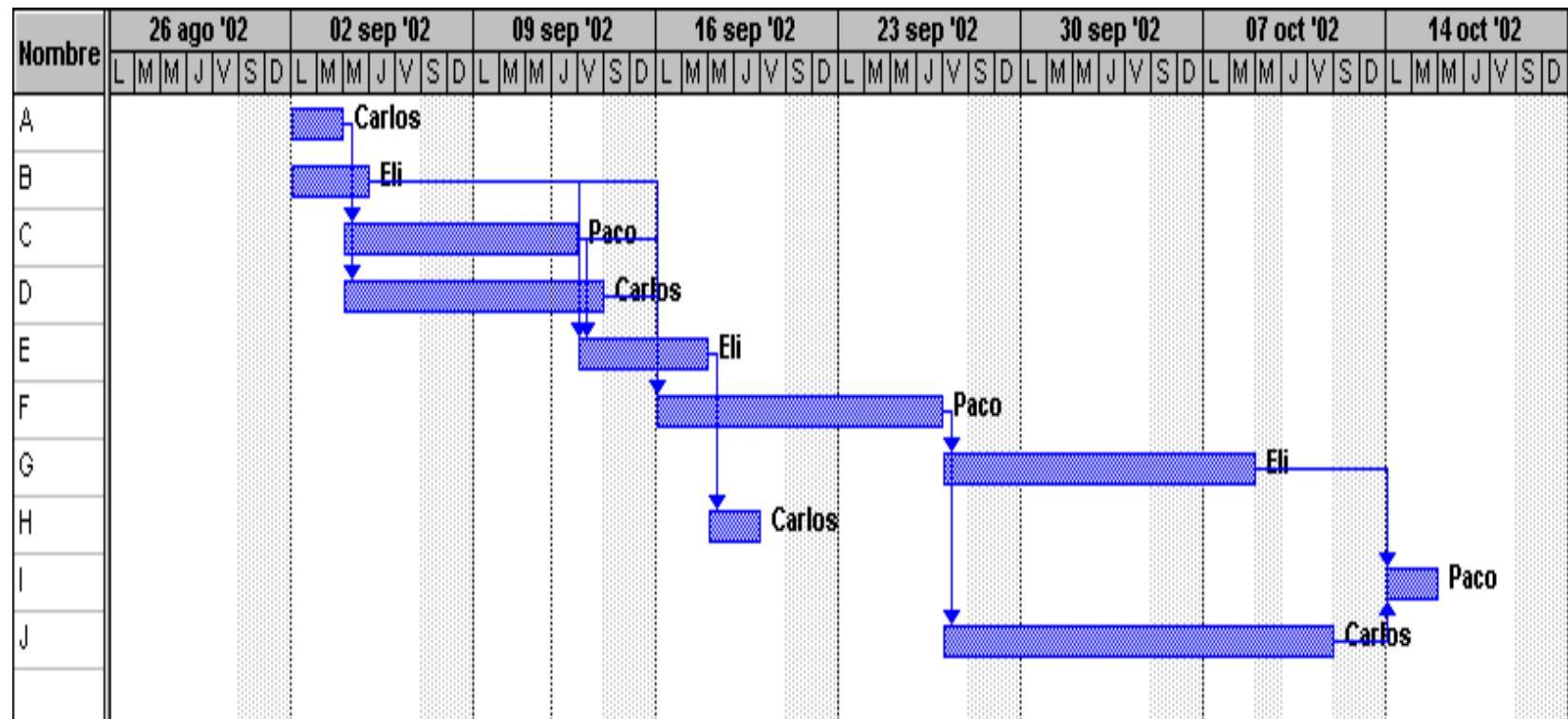
Agenda (II)

Actividades	Fecha inicio	Fecha fin
A		
B		
C		
D		
E		
G		
H		
I		
J		
K		

Diagrama Gant

- Eje de ordenadas
 - Representa Actividades o Recursos
- Eje de abscisas
 - Tiempo
- Permite observar con detalle la evolución del proyecto.
- La herramienta Microsoft Project permite generar automáticamente
 - Diagramas de Gantt
 - Grafos PERT

Diagrama Gant (II)



Ejercicio 1

Tareas	Preced.	Duración.
A	-	2
B	-	3
C	-	2
D	A	3
E	A, B	1
F	A	4
G	B, C	2
H	C	5
N	D, E	3
J	N	2
K	F, J	2
M	F, G	3

Calcular:

1. Tiempo mínimo de duración del proyecto
2. Camino crítico
3. Agenda
4. ¿qué ocurre si G se retrasa en 4 días y M en 3?
5. Si retrasamos G en 1 día, ¿qué le ocurre a M?

Ejercicio 2

Tareas	Preced.	Duración.
A	-	4
B	-	2
C	A , B	1
H	E , C	3
F	B	1
G	B	5
J	F	2
K	F , H	3
L	H , F	6
Q	M , J , G	4
R	Q	5
P	L , K	4
N	K	1
E	B	1
M	H , F	1

Calcular:

1. Tiempo mínimo de duración del proyecto
2. Camino crítico
3. Agenda
4. ¿qué ocurre si J se retrasa en 5 días ?
5. ¿qué ocurre si F se retrasa en 2 días ?

Problemas de *scheduling*

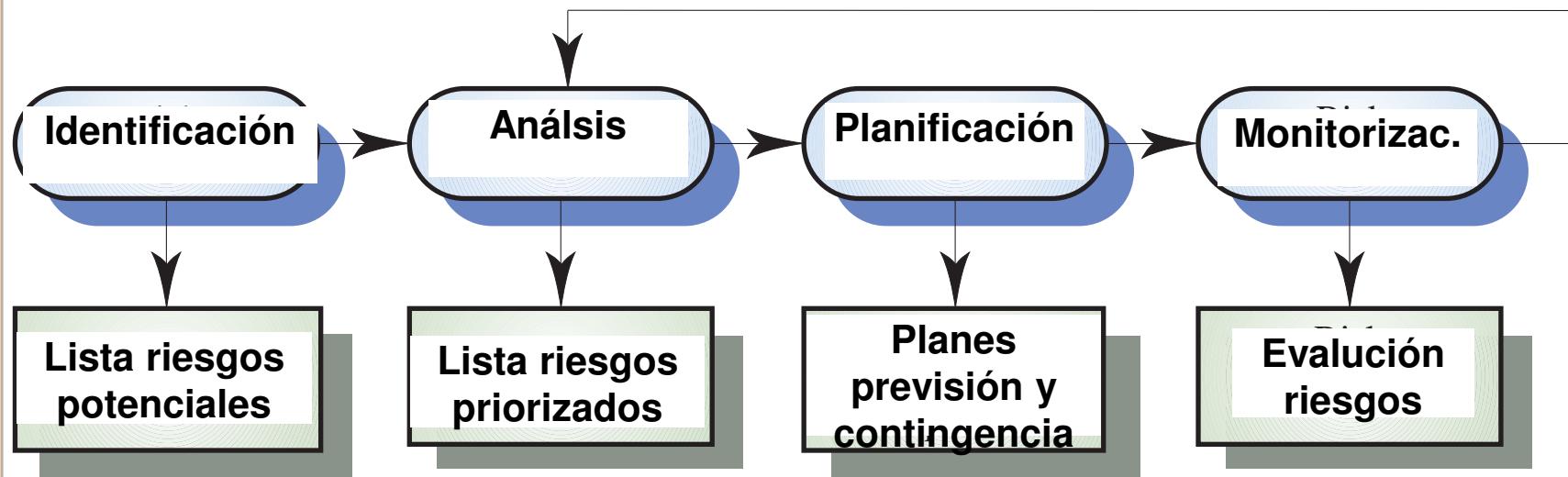
- La tarea de estimar la dificultad del problema y por lo tanto el coste asociado es bastante "dura"
- La productividad no es proporcional al número de gente trabajando en una tarea
- Añadir gente a un proyecto que va con retraso lo retrasará más todavía debido a la sobrecarga en cuanto a las comunicaciones personales
- Lo inesperado SIEMPRE ocurre. Se debe tener esto en cuenta a la hora de planificar

Gestión de riesgos

- Actividad que concierne a la identificación de los riesgos y desarrollo de planes para minimizar sus efectos en un proyecto.
- Un riesgo es una **probabilidad** de que pueda ocurrir alguna circunstancia adversa.
 - Los riesgos del proyecto afectan a la agenda o los recursos
 - Los riesgos del producto afectan a la calidad o realización del desarrollo
 - Los riesgos del negocio afectan a la organización que desarrolla o que gestiona el proyecto

El proceso de gestión de riesgos

- Identificación de riesgos
- Análisis de riesgos
- Planificación de los riesgos
- Monitorización de los riesgos



Identificación de riesgos

TIPO DE RIESGO	POSIBLE RIESGO
TECNOLOGÍA	La Base de Datos utilizada no puede procesar muchas transacciones por segundo como se esperaba.
PERSONAS	Es imposible seleccionar personal con las habilidades requeridas para el proyecto.
ORGANIZACIONAL	Los problemas financieros en la organización causan reducciones en el presupuesto del proyecto.
HERRAMIENTAS	Las herramientas CASE no se pueden integrar.
REQUERIMIENTOS	Se proponen cambios en los requerimientos que suponen rehacer el diseño.
ESTIMACIÓN	El tiempo requerido para desarrollar el software está infraestimado.

Análisis de riesgos

POSIBLE RIESGO	PROBABILIDAD	EFFECTOS
Problemas financieros en la organización.	BAJA	CATASTRÓFICO
Problemas de selección de personal.	ALTA	CATASTRÓFICO
Cambios en los requerimientos.	MODERADA	SERIO
Rendimiento de la Base de Datos.	MODERADA	SERIO
Tiempo de desarrollo infraestimado.	ALTA	SERIO
Herramientas CASE no se pueden integrar.	ALTA	TOLERABLE

Planificación de riesgos

RIESGO	ESTRATEGIA
Problemas financieros en la organización.	Preparar un documento breve para el gestor principal que muestre que el proyecto hace contribuciones muy importantes a los objetivos del negocio.
Problemas de selección de personal.	Alertar al cliente de las dificultades potenciales y las posibilidades de retraso, previsión de cursos de formación.
Cambios en los requerimientos.	Investigar la información para valorar el impacto del cambio en los requerimientos, prototipado y revisión con el cliente.
Rendimiento de la Base de Datos.	Investigar la posibilidad de comprar una base de datos de rendimiento más alto.
Tiempo de desarrollo infraestimado.	Investigar los componentes comprados y la utilización de un generador de programas.

Monitorización de riesgos

TIPO DE RIESGO	IDENTIFICADORES POTENCIALES
TECNOLOGÍA	Entrega retrasada del hardware o del soporte software.
PERSONAS	Personal con moral baja, malas relaciones entre miembros del equipo.
ORGANIZACIONAL	Cotilleos en la organización, pasividad en el gestor principal.
HERRAMIENTAS	Quejas sobre herramientas CASE, peticiones de estaciones de trabajo más potentes.
REQUERIMIENTOS	Peticiones de cambios en muchos requerimientos, quejas del cliente.
ESTIMACIÓN	Fracaso en el cumplimiento de la agenda acordada.

Puntos clave

- Una buena gestión de proyectos es esencial para el éxito del proyecto
- La naturaleza intangible del software causa problemas en la gestión de proyectos
- Los gestores se encargan de varias tareas, pero las más significativas con planificación, estimación y confección de agendas (*scheduling*)
- La planificación y estimación son procesos iterativos y continuados durante todo el desarrollo del proyecto

Tema 4. Gestión de Recursos humanos

-  Límites del pensamiento
-  Introducción
-  Trabajo en grupo
-  Selección y organización del personal
-  El modelo CMM para personal

Bibliografía

- Captítulo 25. Managing people. Software Engineering Sommerville 7^a edición.
- Capítulo 5. Planificación de Proyectos software. Ingeniería del software. 4^a edición. Roger S. Pressman.
- Capítulo 7. Planificación Temporal y seguimiento de proyectos. Ingeniería del software. 4^a edición. Roger S. Pressman.

Introducción (I)

LA GENTE EN EL PROCESO

- La gente es uno de los "bienes máspreciados" de una organización
- Las tareas de un gestor están esencialmente **orientadas a la gente**. A menos que haya algún entendimiento con la gente, la gestión será un fracaso
- La ingeniería del software es fundamentalmente una **actividad cognitiva**. Las limitaciones del conocimiento limitan a su vez el proceso software

Introducción (II)

ACTIVIDADES DE GESTIÓN

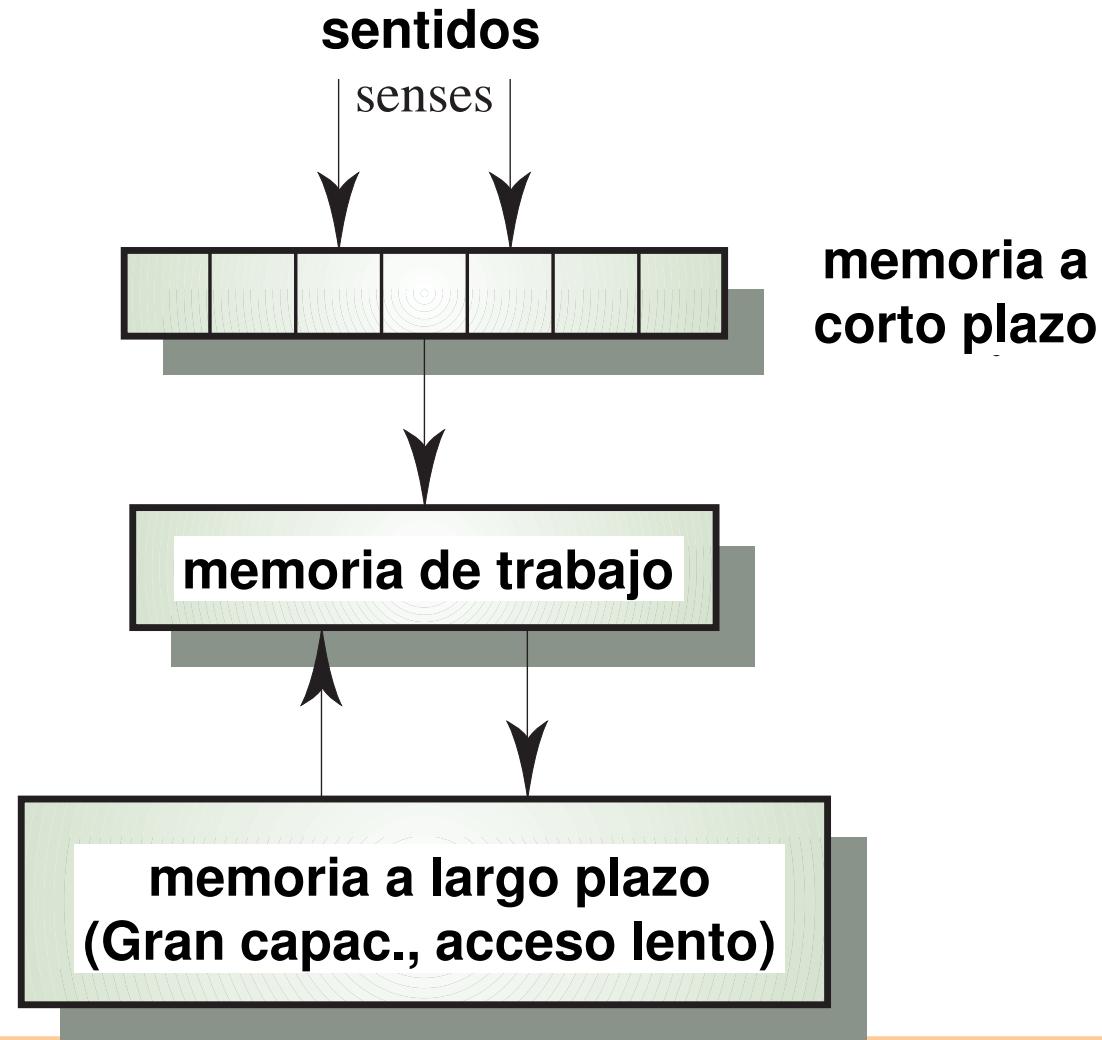
- Solución de problemas (usando la gente disponible)
- Motivación (de la gente que trabaja en el proyecto)
- Planificación (QUÉ tiene que hacer la gente)
- Estimación (CUÁN rápido tiene que trabajar la gente)
- Control (de las actividades de la gente)
- Organización (CÓMO tiene que trabajar la gente)

Límites del pensamiento

- La gente no piensa de la misma manera, pero cada uno está sujeto a restricciones básicas sobre su forma de pensar debido a:
 - La organización de la memoria
 - La representación del conocimiento
 - La influencia de la motivación

- Si comprendemos estas restricciones, podemos comprender cómo éstas afectan a la gente que participa en un proceso software

Organización de la memoria



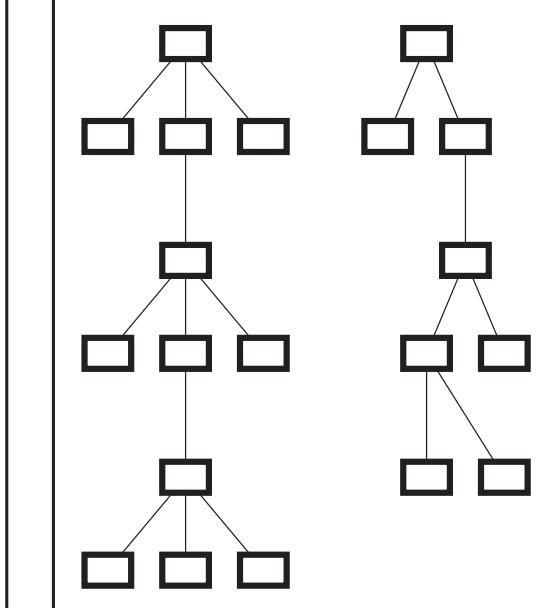
Transferencia de información

- La resolución de problemas requiere normalmente una transferencia de información entre la memoria a corto plazo y la memoria de trabajo
- La información puede perderse o corromperse durante esta transferencia
- El procesamiento de la información ocurre durante la transferencia de información desde la memoria a corto plazo a la memoria a largo plazo

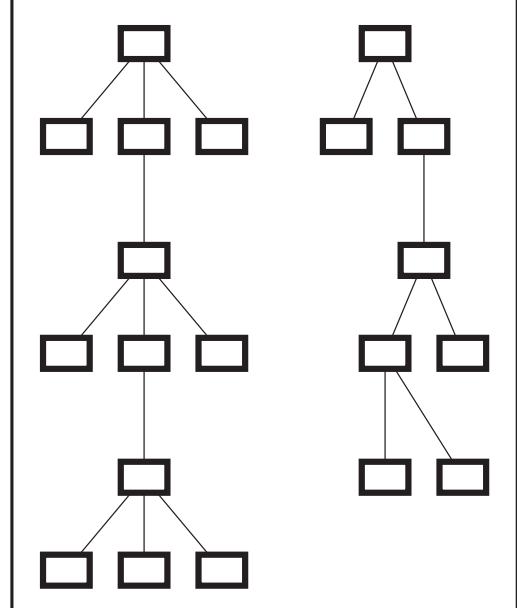
Modelado del conocimiento

- **Conocimiento semántico:** es el conocimiento de los conceptos tales como la operación de asignación, el paso de parámetros, etc.
- **Conocimiento sintáctico:** se refiere a los detalles de representación, por ejemplo un bucle en C.
- El conocimiento semántico se almacena de forma estructurada, independientemente de la representación.

Conocim. sintáctico y semántico

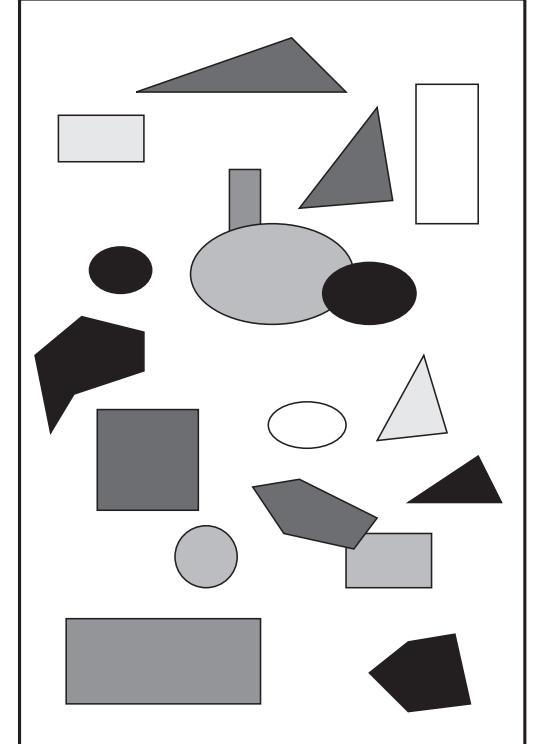


Conocim. de tareas



Conocim. del ordenador

Conocim. semántico



Conocim. sintáctico

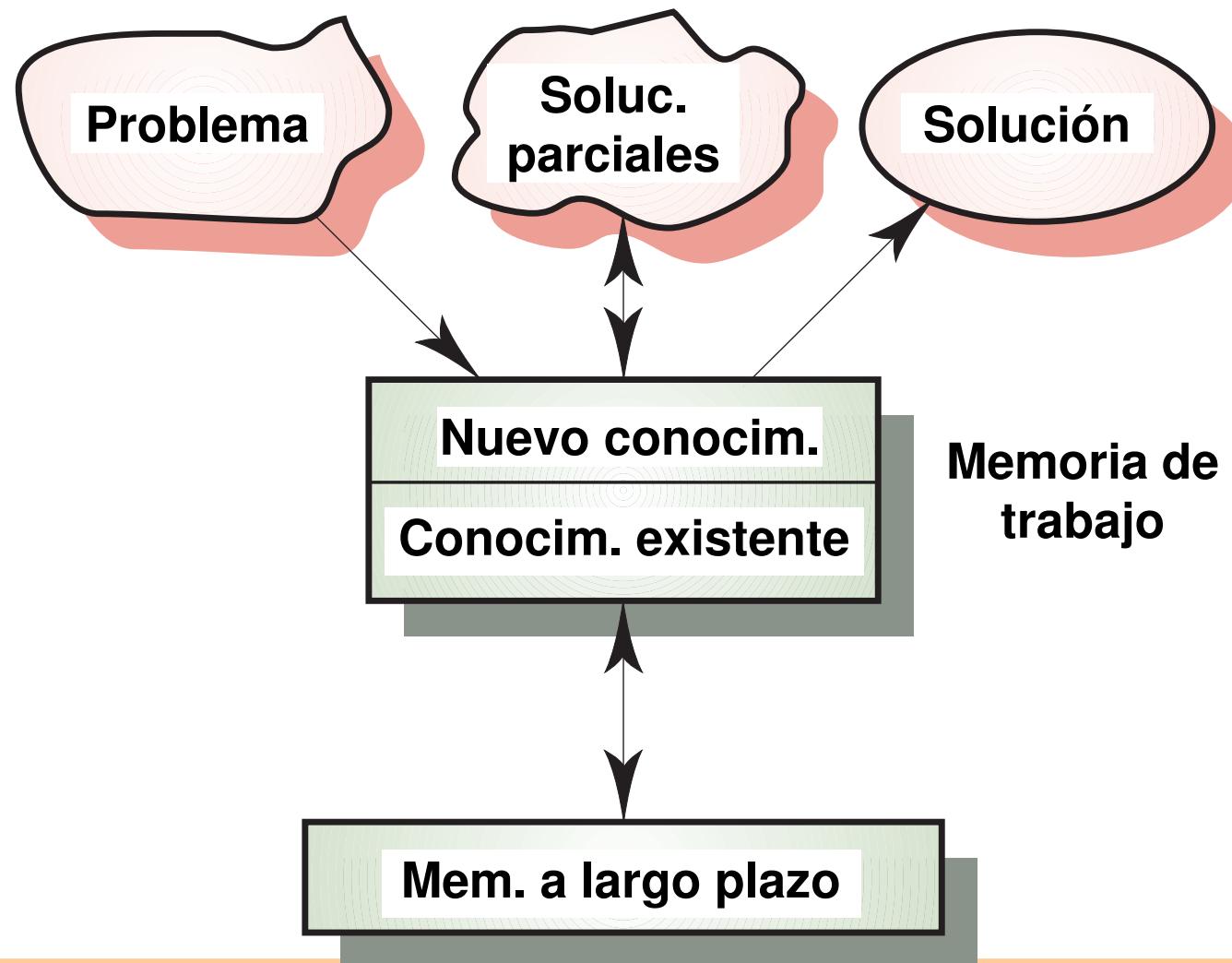
Adquisición del conocimiento

- El conocimiento semántico se adquiere mediante experiencia y aprendizaje activo
- El conocimiento sintáctico se adquiere mediante memorización.
- El nuevo conocimiento sintáctico puede interferir con conocimiento sintáctico ya existente.
 - Suele haber problemas con programadores experimentados en cuanto a mezcla de sintaxis entre diferentes lenguajes de programación

Resolución de problemas (I)

- Es independiente del lenguaje de programación
- Requiere la **integración** de diferentes tipos de conocimiento (del ordenador, de tareas, del dominio, de la organización)
- Implica el desarrollo de un modelo **semántico** de la solución y la **prueba** de dicho modelo contrastándolo con el problema
- Conduce a la representación del modelo en una notación adecuada o lenguaje de programación

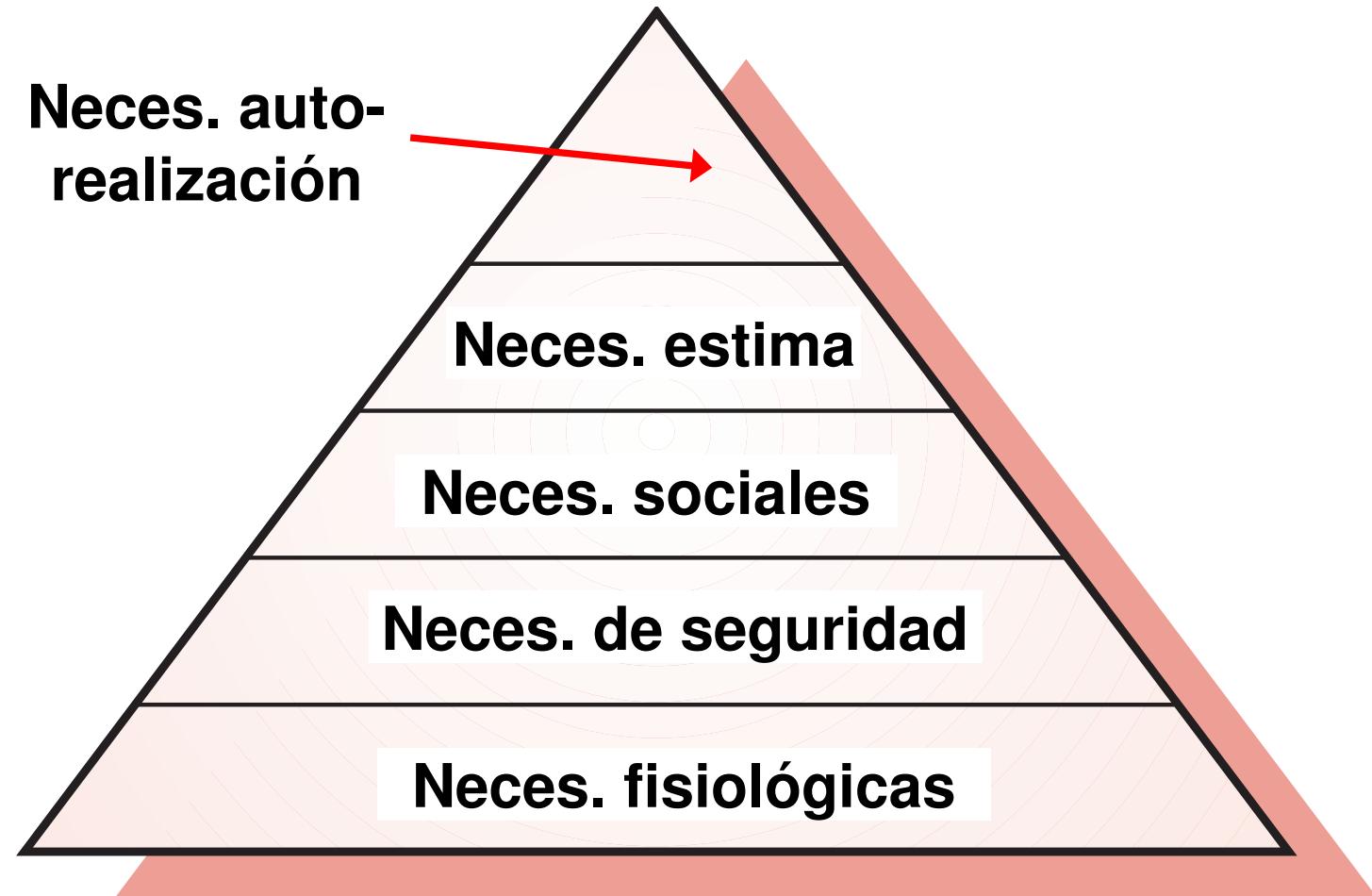
Resolución de problemas (II)



Motivación

- Una tarea importante de un gestor es la de motivar a la gente que trabaja en un proyecto
- La motivación es una tarea compleja. Se considera que hay diferentes tipos de motivación basadas en:
 - Necesidades básicas (ej. comer, dormir, etc.)
 - Necesidades personales (ej. respeto, autoestima)
 - Necesidades sociales (ej. ser aceptado como parte de un grupo)

Jerarquía de necesidades



Tipos de personalidad

- La jerarquía de necesidades es realmente una simplificación de la realidad
- La motivación debería tener en cuenta también los diferentes tipos de personalidad:
 - Orientados a la tarea
 - Orientados a sí mismos
 - Orientados a la interacción

Aspectos sobre motivación

- La motivación se consigue en la medida en que se puedan satisfacer las necesidades de un individuo
- La motivación puede cambiar dependiendo de circunstancias externas al personal o eventos externos
- La gente no solamente está motivada por factores personales, sino por los derivados de formar parte de un grupo y una cultura.

El trabajo en grupo

- La mayor parte del trabajo de ingeniería del software es una actividad de grupo
 - El desarrollo de la agenda para la mayor parte de proyectos no triviales es tal que no puede completarse por una persona trabajando en solitario
- La **interacción** con el grupo es una clave determinante del rendimiento del grupo
- La flexibilidad en la **composición** de un grupo es limitada
 - Los gestores deben intentar trabajar lo mejor posible con la gente disponible

Distribución del tiempo



Composición del grupo

- Un grupo formado por miembros que comparten la misma motivación puede ser problemático
 - Orientados a la tarea - cada uno quiere hacer las cosas según su propio criterio
 - Orientados a sí mismo - cada uno quiere ser el jefe
 - Orientados a la interacción - demasiadas "charlas", no suficiente trabajo
- Un grupo efectivo tiene un equilibrio de todos los tipos
- La mayoría de los ingenieros son orientados a la tarea
- Necesidad de que todos los miembros se impliquen en las decisiones que afecten al grupo

El líder del grupo

- El liderazgo del grupo se debe basar en el **respeto**, no en un título que proporciona un "status"
- Debe haber un líder administrativo y técnico
- Un liderazgo democrático es más efectivo que uno autocrático
- Se debería soportar una trayectoria de carrera profesional basada en la competencia técnica

Cohesión del grupo

- En un grupo cohesivo, los miembros consideran que el grupo es más importante que un individuo del mismo
- Ventajas de un grupo cohesivo:
 - Se pueden desarrollar estándares de calidad del grupo
 - Los miembros del grupo trabajan estrechamente, por lo que se reducen las inhibiciones causadas por la ignorancia
 - Los miembros del grupo aprenden unos de otros y dan a conocer su trabajo entre sus miembros
 - Se puede practicar la "programación sin ego", en la que los miembros se esfuerzan por mejorar el trabajo de los demás

Desarrollo de la cohesividad

- La cohesividad está influenciada por factores tales como la cultura organizacional y las personalidades del grupo
- La cohesividad se puede propiciar mediante
 - Eventos sociales
 - Desarrollar una identidad de grupo y un área propia
 - Actividades explícitas de construcción de grupos
- La sinceridad con la información es una forma sencilla de asegurar que todos los miembros se sientan parte del grupo

Comunicaciones del grupo

- Una buena comunicación es esencial para el trabajo efectivo del grupo
- Se debe intercambiarse información sobre el estado del trabajo, las decisiones de diseño y los cambios en las decisiones previas
- Una buena comunicación fortalece la cohesividad del grupo y promueve un mayor entendimiento
- Factores que influyen:
 - Status de sus miembros
 - Personalidades de sus miembros
 - Composición sexual del grupo
 - Canales de comunicación

Organización del grupo

- Tamaño del grupo relativamente pequeño (menos de ocho personas)
- Dividir los proyectos grandes en múltiples proyectos pequeños
- Los grupos pequeños pueden organizarse de forma informal y democrática
- El jefe de programadores intentará hacer un uso efectivo de las habilidades y experiencia del grupo

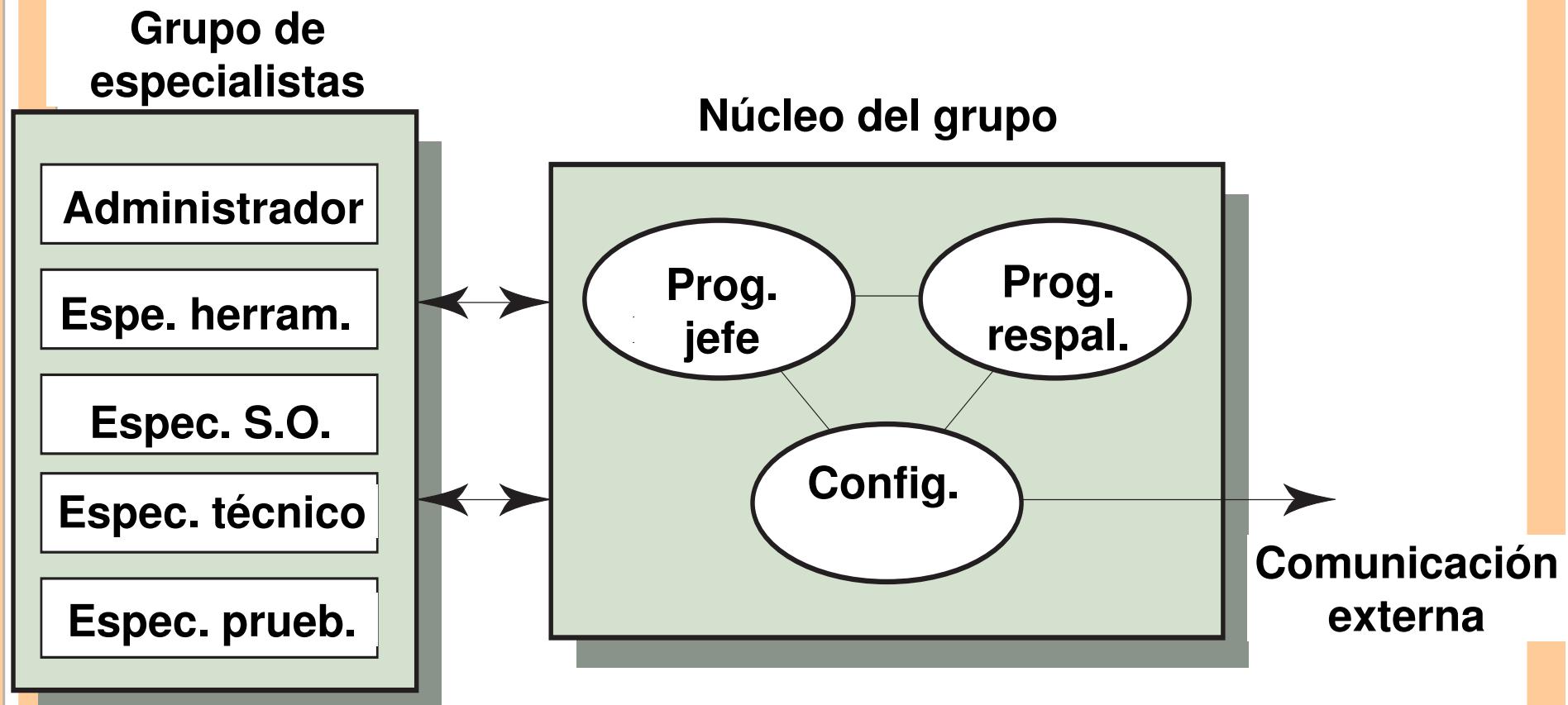
Organización democrática

- El grupo actúa como un todo y las decisiones se toman por consenso
- El líder del grupo sirve como un enlace externo del grupo, pero no realiza asignaciones específicas de trabajo
- El trabajo se discute por el grupo como un todo y las tareas se reparten de acuerdo según la habilidad y experiencia de cada uno
- Esta aproximación tiene éxito en grupos cuyos miembros son todos competentes y con experiencia

Grupos de programac. extrema

- Los grupos de "programación extrema" son una variante de la organización democrática
- En dichos grupos, se toman algunas decisiones de gestión por parte de los miembros del grupo
- Los programadores trabajan por parejas y adquieren una responsabilidad colectiva del código que han desarrollado

Grupos con jefe de trabajo



Problemas

- Los buenos diseñadores y programadores no se encuentran fácilmente.
- Los restantes miembros del grupo pueden verse afectados por el hecho de que el jefe de programadores asuma el éxito de todo el grupo, dificultando su trabajo
- Debido a la estructura impuesta por la organización, puede no ser posible formar este tipo de grupo

Selección y organiz. de personal

- Se trata de una responsabilidad importante del gestor de proyectos
- Normalmente las decisiones se basan en:
 - información proporcionada por el candidato (su currículum)
 - información obtenida mediante una entrevista
 - recomendaciones de otra gente que conoce al candidato
- Algunas compañías utilizan test psicológicos y/o test de aptitud
 - No hay evidencias de si estos tests son o no son realmente útiles

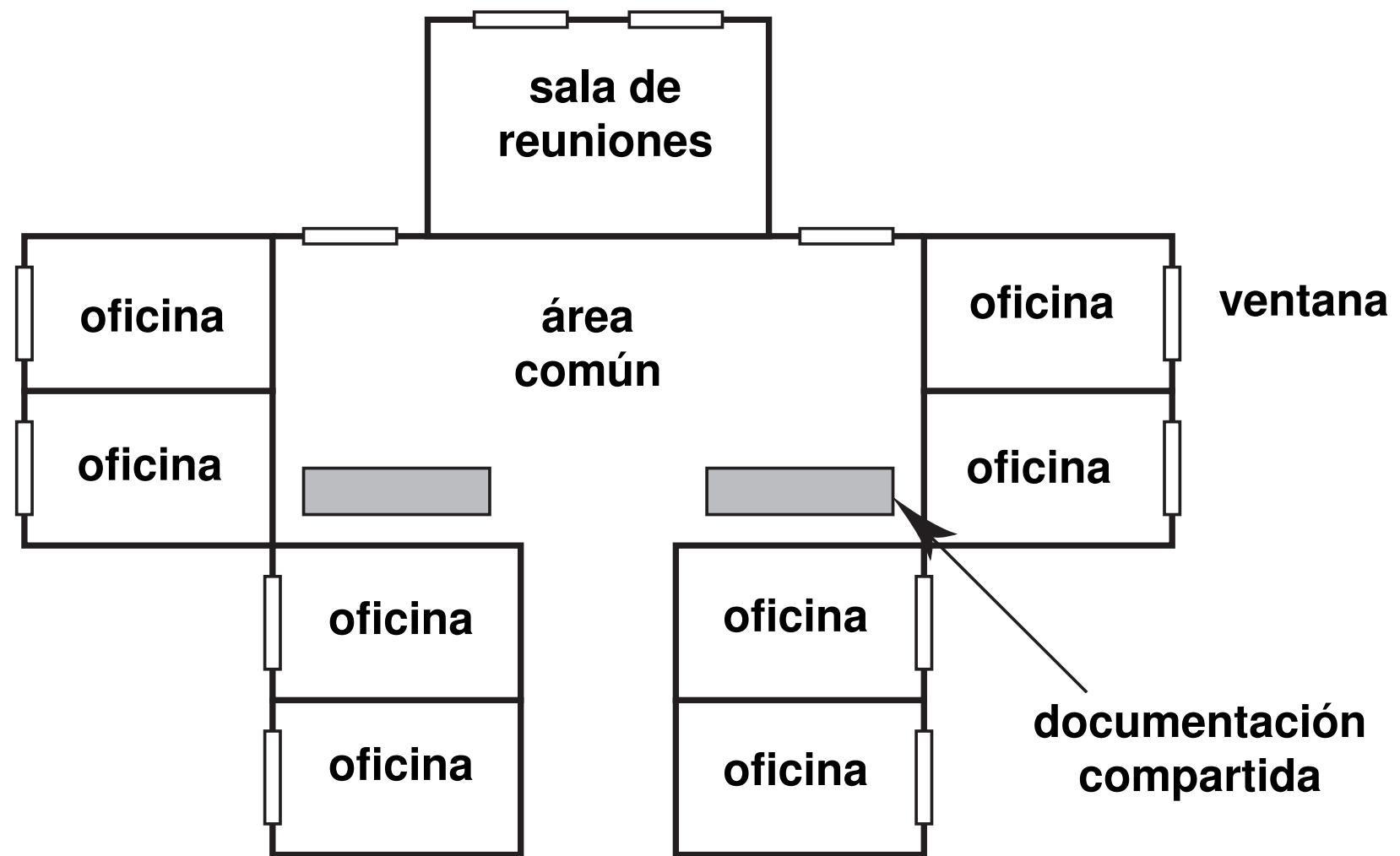
Factores de selección

- Experiencia en el dominio de la aplicación
- Experiencia en la plataforma
- Experiencia en el lenguaje de programación
- Estudios previos
- Capacidad de comunicación
- Adaptabilidad
- Actitud
- Personalidad

Entornos de trabajo

- El entorno físico de trabajo juega un papel importante en la productividad y satisfacción individual
 - Confort
 - Privacidad
 - Otras facilidades
- Las consideraciones acerca de la salud y la seguridad deberían tenerse en cuenta
 - Iluminación
 - Climatización
 - Mobiliario

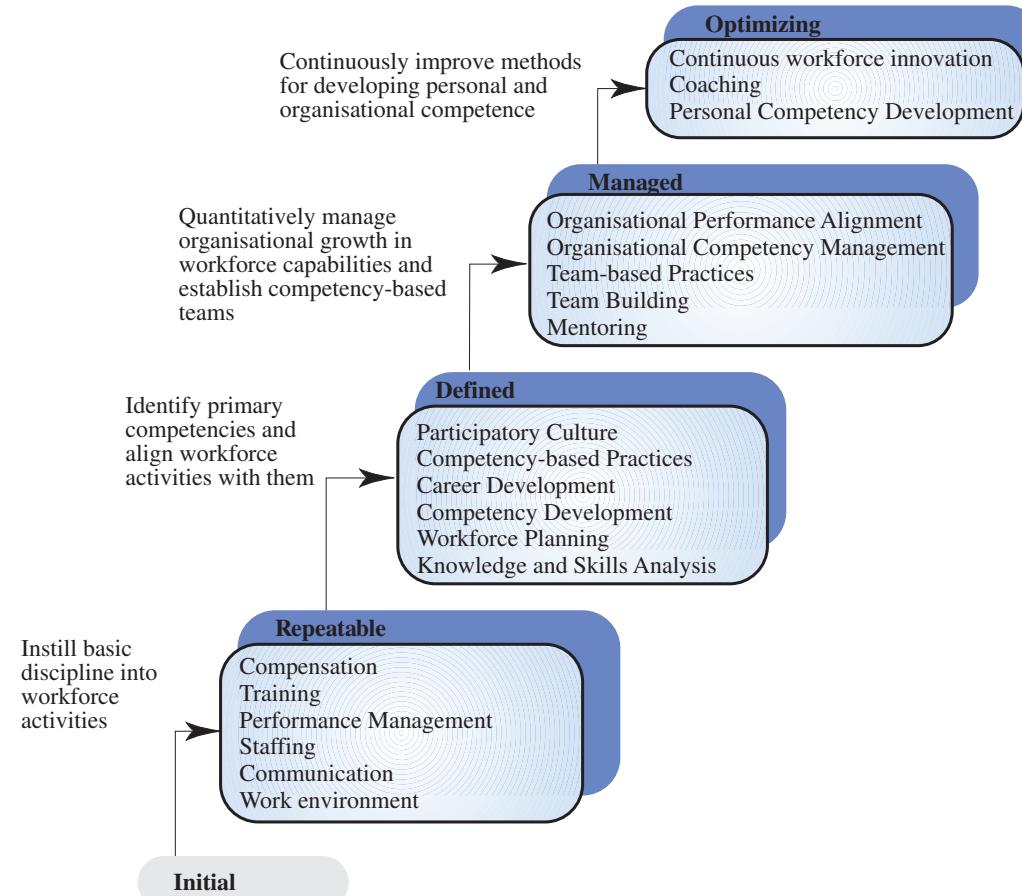
Oficina hipotética



El modelo CMM de personal

- Pretende ser un marco para la **gestión** del trabajo realizado por la gente implicada en el desarrollo del software
- Es un modelo de **cinco** etapas
 - Inicial. Gestión de rec. humanos "ad-hoc"
 - Repetible. Se desarrollan políticas para mejora de las capacidades (aptitudes)
 - Definido. Gestión de rec. humanos estandarizada para la organización
 - Gestionado. Se establecen metas cuantitativas para la gestión de recursos humanos
 - Optimizado. Se realiza un esfuerzo continuado para mejorar la competencia y motivación en el trabajo

El modelo CMM de personal



Objetivos P-CMM

- Mejorar las capacidades de la organización mejorando las capacidades de trabajo de la gente implicada
- Asegurar que las capacidades para el desarrollo del software no conciernen a un número pequeño de individuos
- Igualar la motivación de los individuos con la de la organización
- Ayudar a la "retención" de gente con conocimientos y habilidades críticas

Puntos clave

- Los gestores de software deben comprender algunos de los factores humanos para poder llevar a cabo su trabajo con éxito
- Los factores principales a considerar son:
 - La organización de la memoria
 - La representación del conocimiento
 - La influencia de la motivación
- La composición y comunicación de los grupos de trabajo resulta fundamental para el éxito del proyecto
- El modelo P-CMM proporciona un marco para mejorar las capacidades de los recursos humanos de una organización

Tema 5.

Gestión de la Configuración del Software

-  Definición
-  Identificación
-  Control de versiones
-  Control de cambios
-  Auditorías e informes
-  Construcción del sistema

Bibliografía

Captítulo 29. Gestión de configuraciones. Software Engineering Sommerville 7^a edición.

Capítulo 9. Gestión de configuraciones. Ingeniería del software. 5^a edición. Roger S. Pressman.

Definición

Actividad de autoprotección para:

- Identificar el cambio
- Controlar el cambio
- Garantizar que el cambio se implementa adecuadamente
- Informar del cambio a todos aquellos que les interese

Gestión de configuraciones ≠ Mantenimiento

Planificación de la GC

- Definición de entidades a gestionar y esquema formal de identificación
- Identificación de responsables de los procedimientos de la GC
- Descripción de cómo los registros de documentos del proceso de GC deberían ser mantenidas
- Descripción de las herramientas usadas
- Definición de la base de datos de configuración (BDC) usada para almacenar la información

Identificación de EC

Los elementos de configuración son aquellos documentos que se van a requerir para un futuro mantenimiento

Nombrado jerárquico:

- Asociado a proyectos particulares
- No reusable

No debería cambiar de forma arbitraria

Base de datos de configuraciones (I)

Usada para registrar cualquier información relevante relacionada con las configuraciones

Sirve de apoyo a la evaluación del impacto del cambio

Debe proporcionar respuestas a preguntas cómo:

- (1) ¿A qué clientes se les ha entregado una versión particular del sistema?
- (2) ¿Qué hw y SO son necesarios para ejecutar una determinada versión?
- (3) ¿Cuántas versiones del sistema se han creado y cuáles son sus fechas de creación?

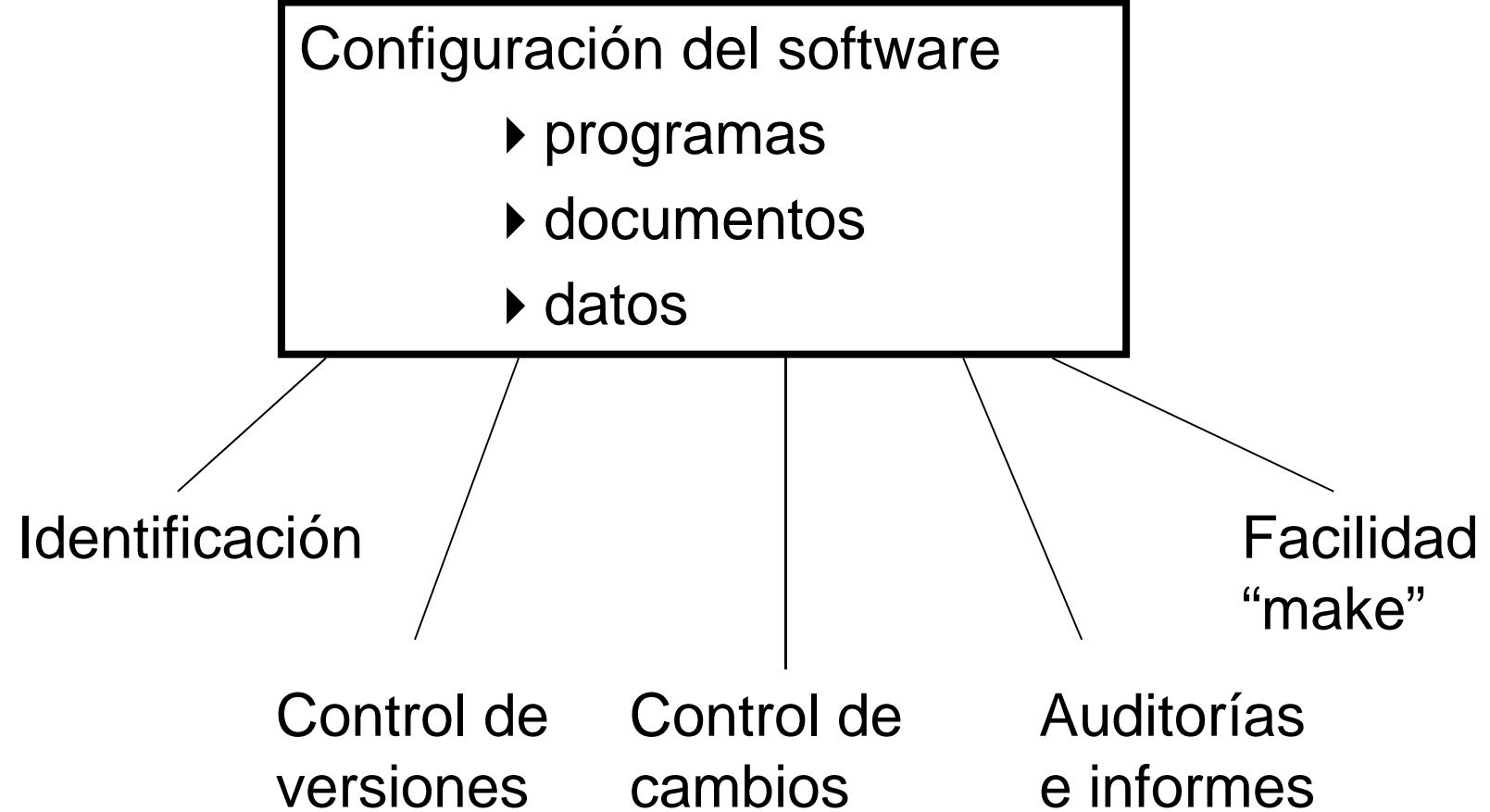
Base de datos de configuraciones (II)

- (4) ¿Qué versiones de un sistema podrían verse afectadas si una componente particular es cambiada?
- (5) ¿Cuántas peticiones de cambio se han hecho sobre una determinada versión?
- (6) ¿Cuántos fallos se han registrado para una versión particular?

Puede ser creada como:

- Un sistema separado
- Integrada con la gestión de versiones y sistema de control que almacena los documentos formales del proyecto

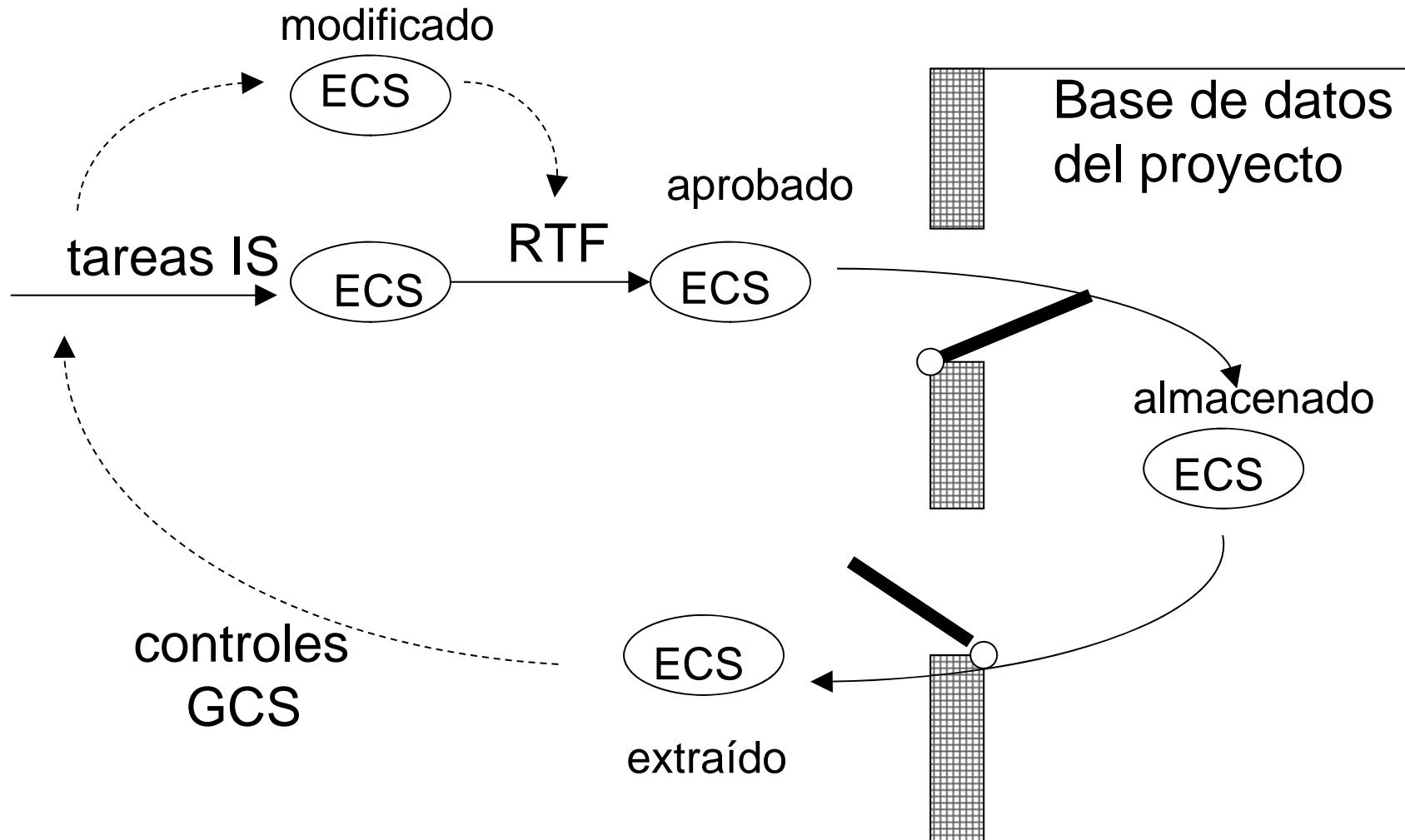
Gestión de configuraciones



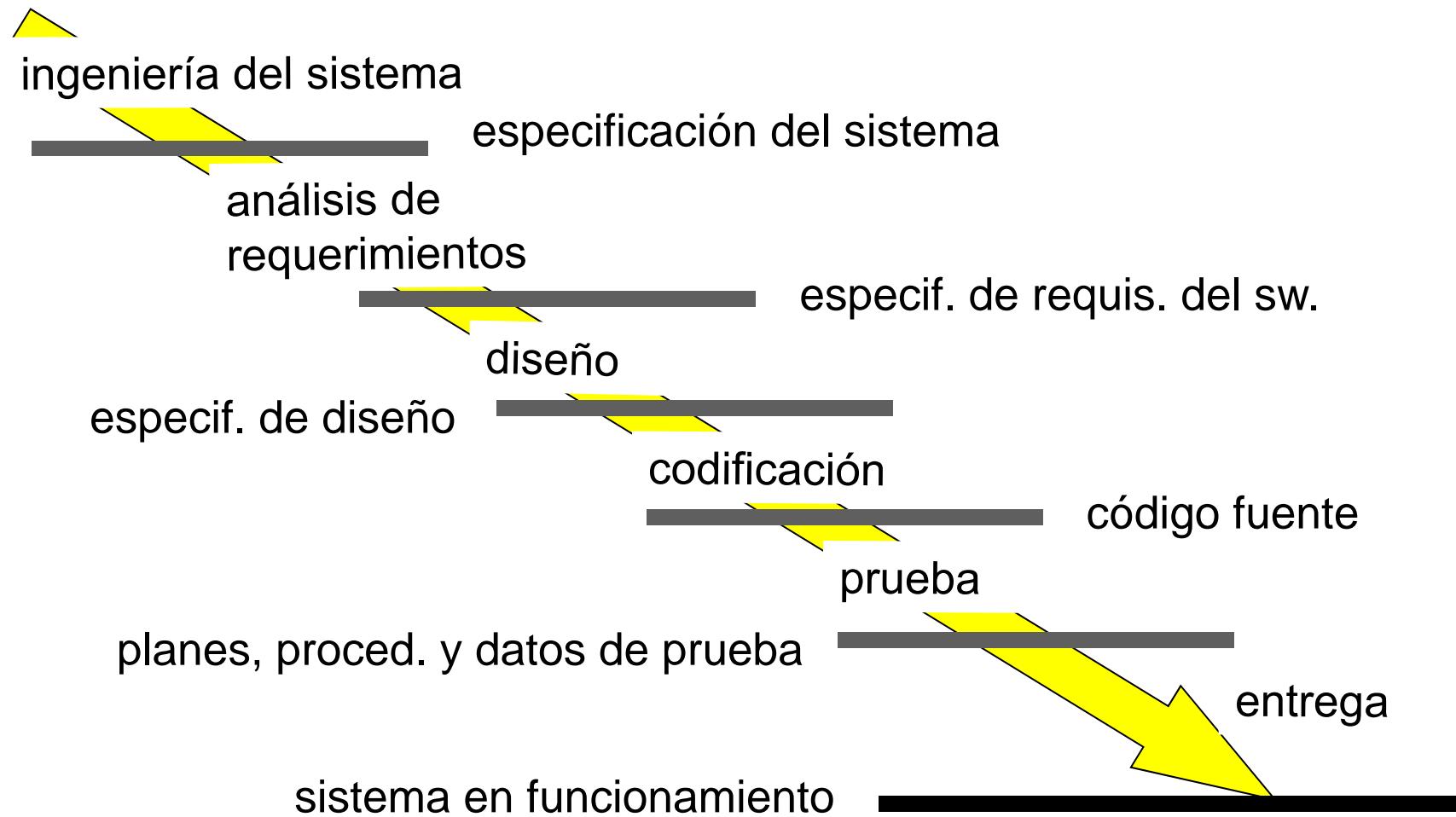
ECS'S típicos

-  Especificación del sistema
-  Plan del proyecto software
-  Especificación de requerimientos del software
-  Manual de usuario preliminar
-  Especificación de diseño
-  Diseño preliminar y detallado
-  Listados de código fuente
-  Planes y procedimientos de prueba
-  Manual de instalación, operación y usuario
-  Ejecutables software

Líneas base (I)



Líneas base (II)



Control de versiones (I)

Implica la identificación y seguimiento de diferentes versiones y “*releases*” del sistema

- Diseño de procedimientos para asegurar que diferentes versiones del sistema puedan recuperarse cuando se requieran, y evitar que no sean cambiadas accidentalmente
- Hay un contacto con el cliente para planificar cuándo deberían distribuirse nuevas *releases*

Control de versiones (II)

- VERSION:** instancia de un sistema que difiere de algún modo de otras instancias (funcionalidad, rendimiento, fallos...)
- RELEASE:** es una versión del sistema distribuida a los clientes (funcionalidades nuevas o nueva plataforma)
- Incluye:
- Ficheros de configuración
 - Ficheros de datos
 - Instalación del programa
 - Documentación electrónica y sobre papel describiendo el sistema

Control de versiones (III)

Nueva versión ➔ nuevo fuente + construcc. stma. (1)

Nueva release ➔ (1) + fich. datos y conf + nueva docum.

El grupo de GC debe decidir cuándo los componentes afectados por un cambio deben ser reconstruidos en una nueva versión o una nueva release

Decisión forzada a veces por fallos descubiertos por el cliente
➔ parcheo del código objeto (mejor solución: versión nueva sin documentación)

Herramientas de gest. de versiones

Unix : SCCS, RCS

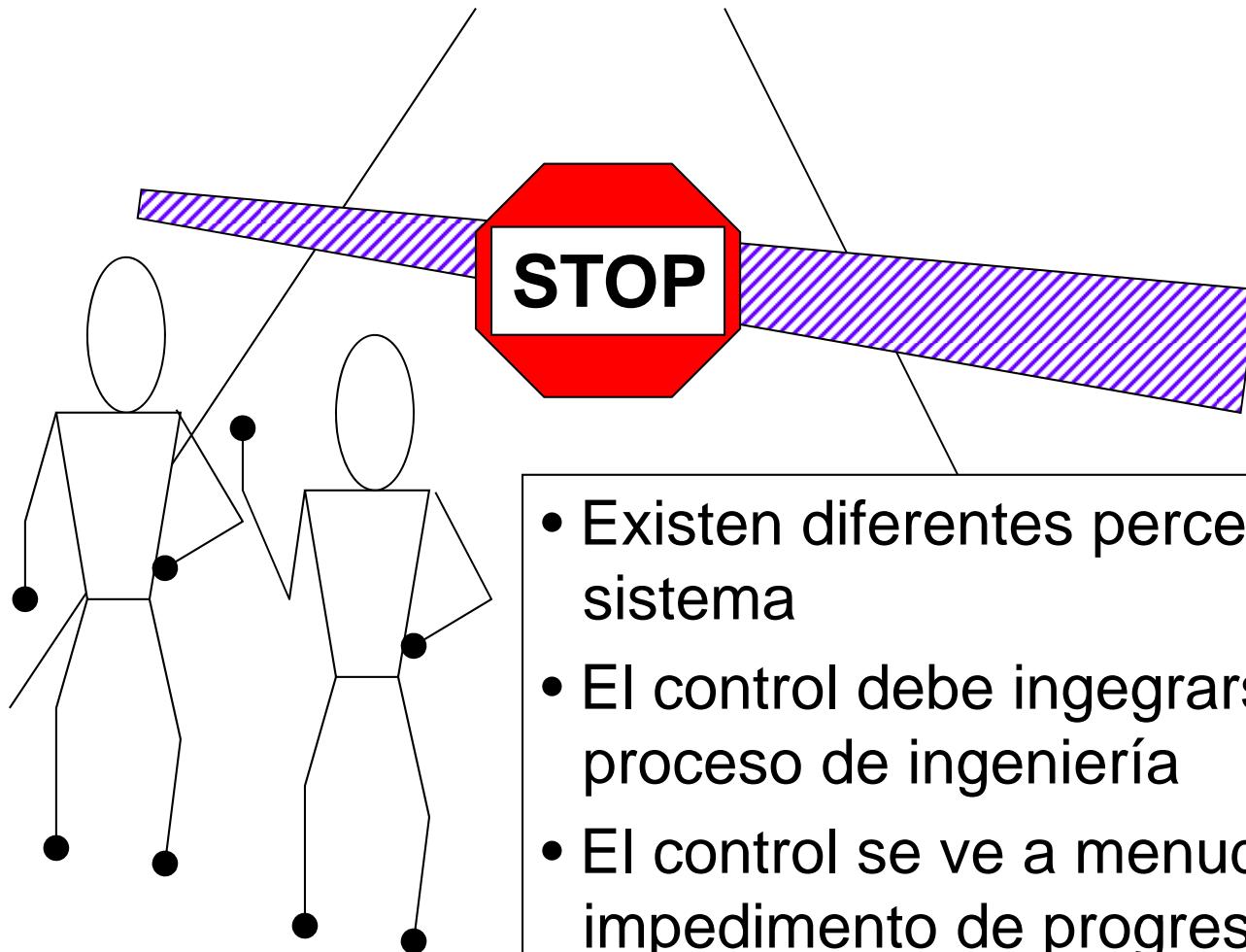
- Identificación de versiones y releases
- Cambios controlados
- Gestión de almacenamiento
- Registro de la historia de cambios

RCS:

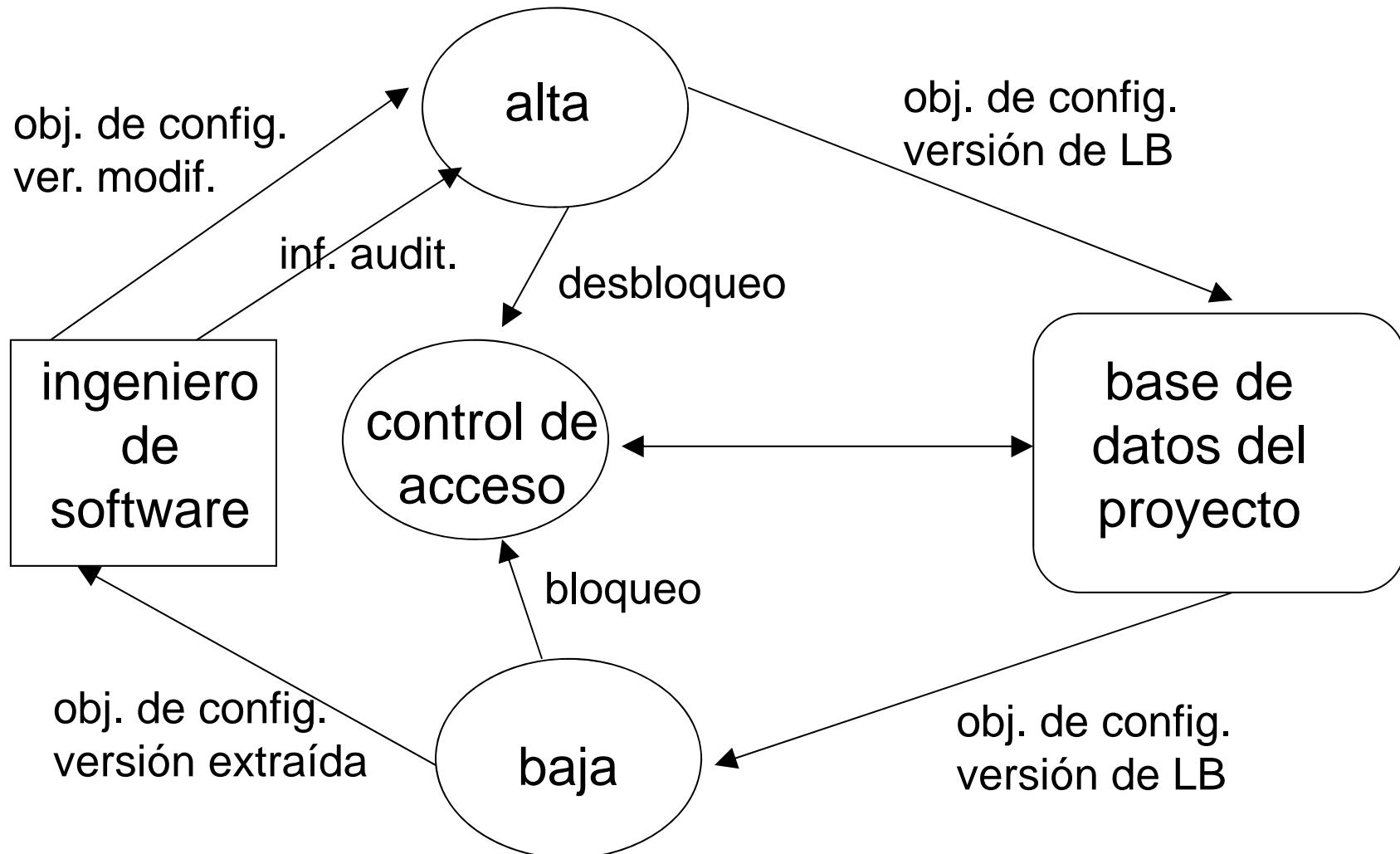
- Graba el código fuente de la versión más reciente
- Soporta el desarrollo paralelo de diferentes releases
- Capacidad de “mezcla” de versiones

Diseñados para trabajar con texto ASCII

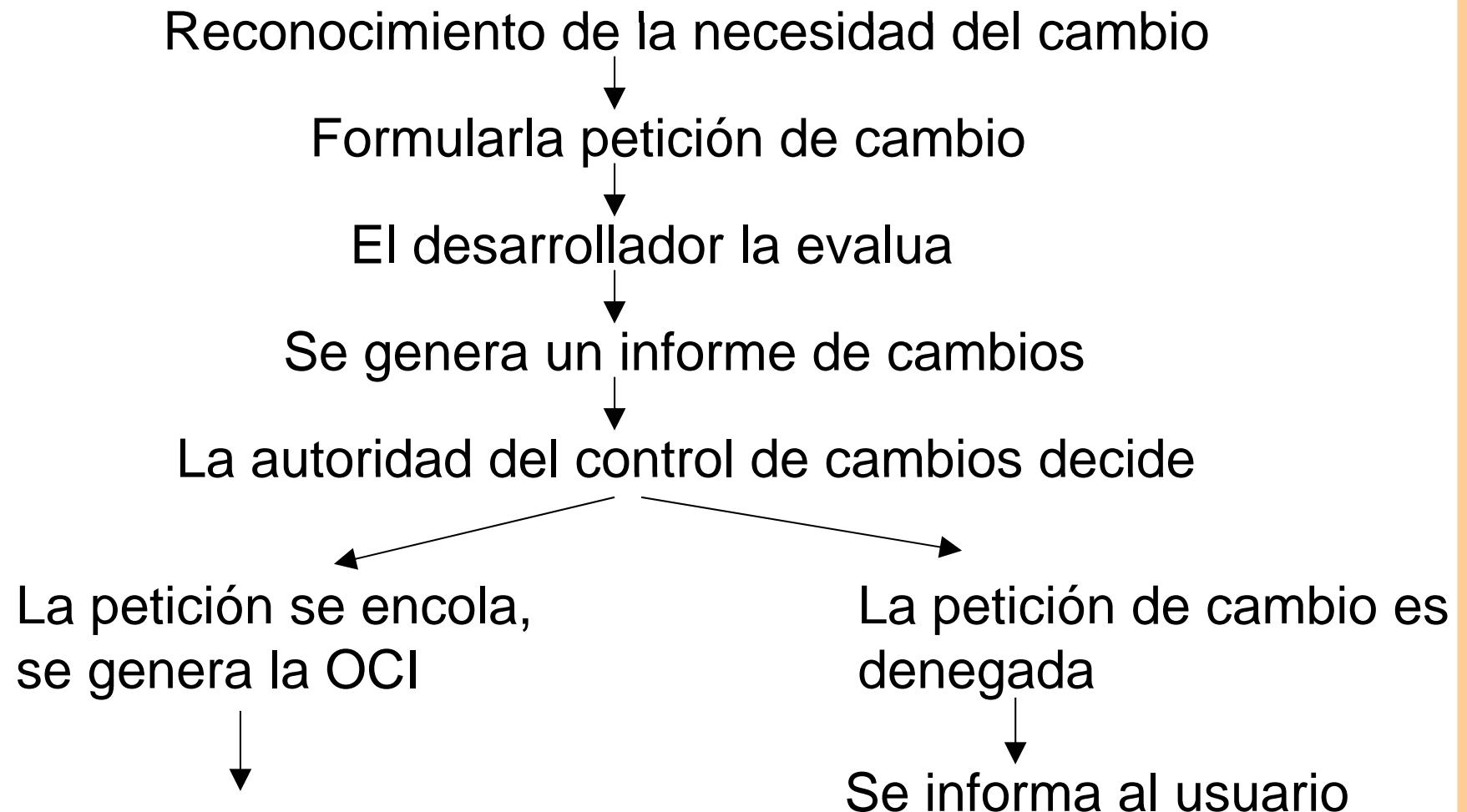
Control de Cambios



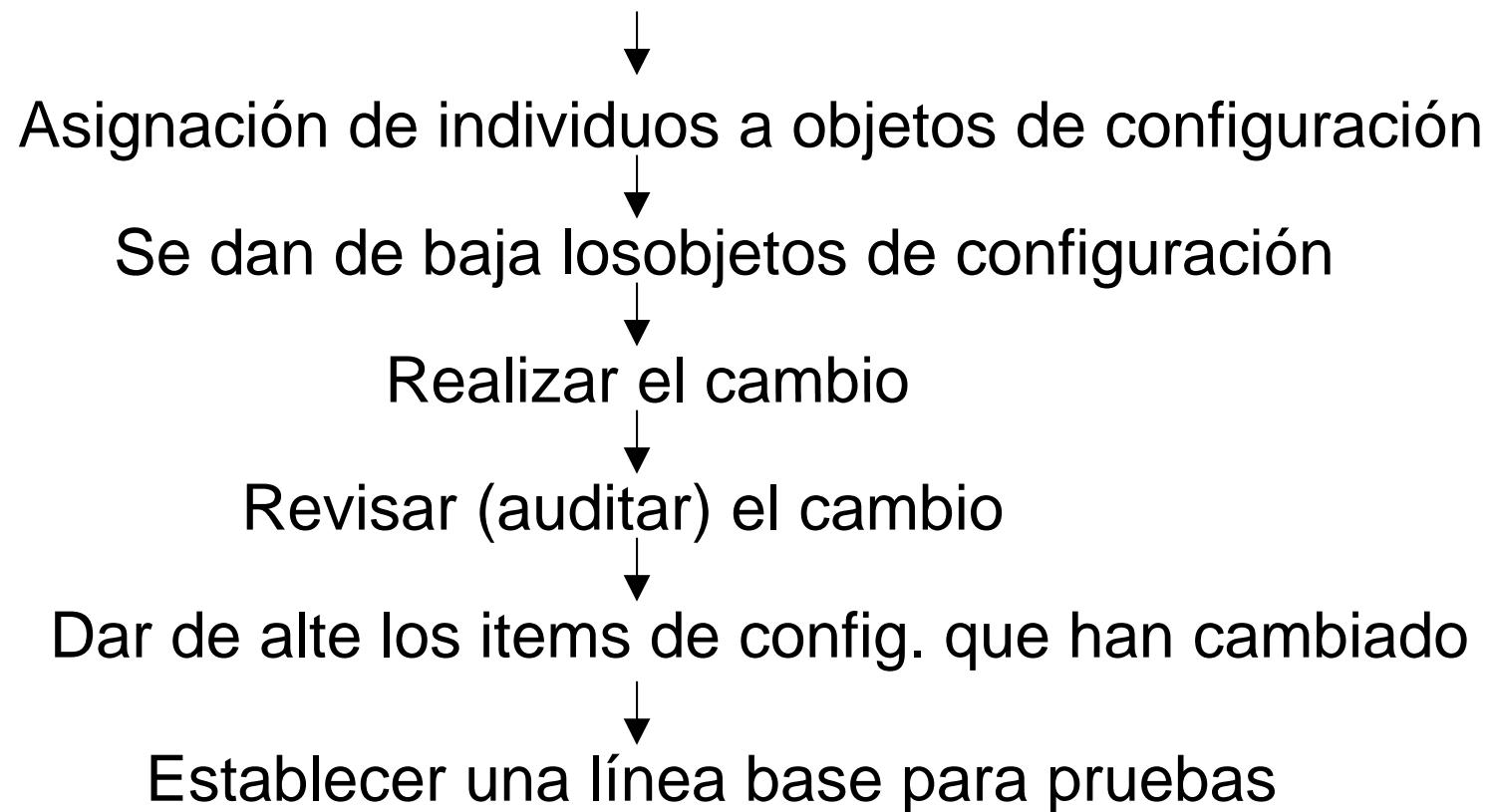
Control de Acceso



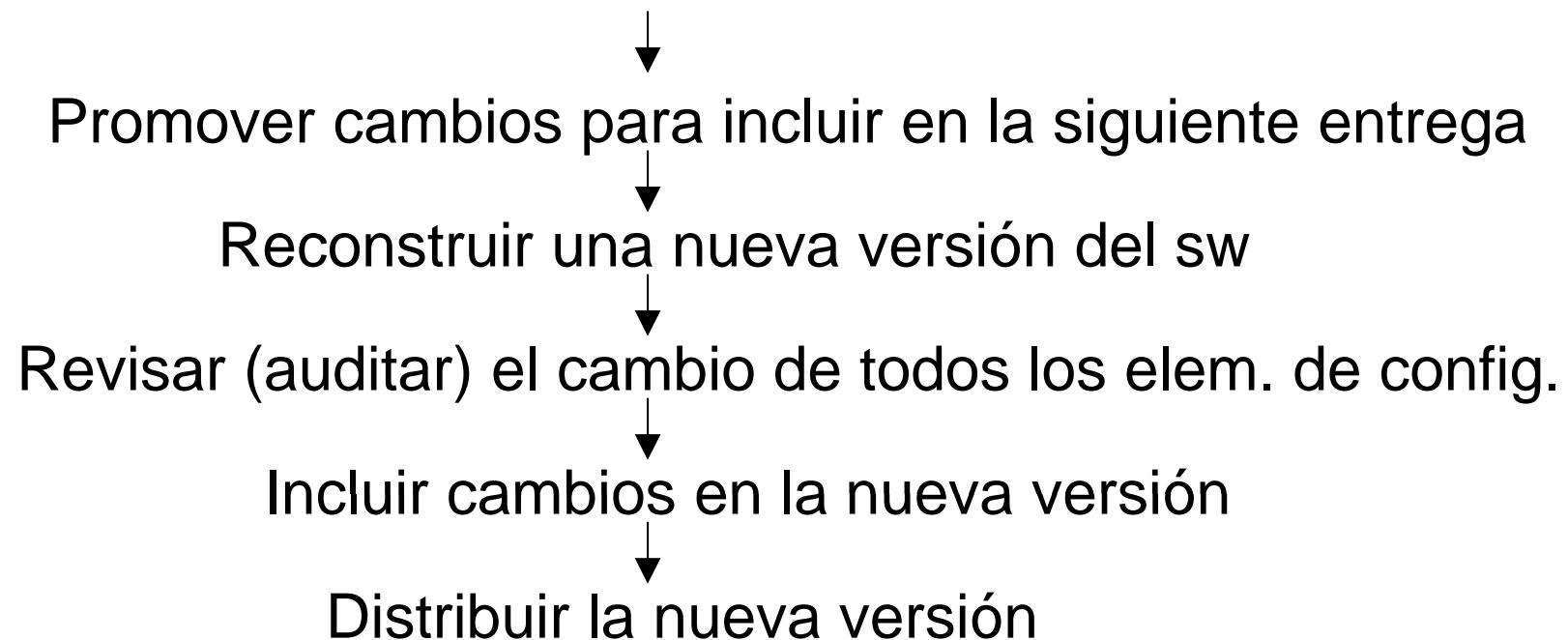
Proceso de control de cambios (I)



Proceso de control de cambios (II)



Proceso de control de cambios (III)



Auditoría de la Configuración

¿Cómo podemos asegurar que el cambio se ha implementado correctamente?

- ❑ Revisiones técnicas formales
 - «corrección técnica»
- ❑ Auditorías de configuración
 - «responde a las siguientes preguntas:
 - ❑ ¿se ha hecho el cambio especificado?
 - ❑ ¿se han seguido los estándares de IS?
 - ❑ ¿se han seguido procedimientos para señalar el cambio, registrarlo y divulgarlo?
 - ❑ ¿se han actualizado adecuadamente todos los ECS relacionados? »

Informes de estado

Responde a las siguientes preguntas:

- ❑ ¿Qué pasó?
- ❑ ¿Quién lo hizo?
- ❑ ¿Cuándo pasó?
- ❑ ¿Qué más se vió afectado?

Construcción del sistema (I)

Proceso de combinar componentes en un programa que se ejecuta sobre una configuración destino particular

Implica:

- Compilación
- Linkado

Hay que llevar especial cuidado con sistemas desarrollados con un sistema distinto del de destino.

Construcción del sistema (II)

Factores a considerar:

- ¿Han sido incluidas todas las componentes?
- ¿Tienen la versión adecuada?
- ¿Están disponibles todos los ficheros de datos?
- ¿Los datos tienen el mismo nombre en la componente y en la máquina destino?
- ¿La versión del compilador es la adecuada?

El proceso se refiere normalmente a componentes físicos.

Herram. de construcc. del sistema

- La herramienta más ampliamente usada para sistemas UNIX es MAKE.
- Mantiene una correspondencia entre el código fuente y las versiones de código objeto de un sistema
- El usuario especifica las dependencias de los componentes y MAKE fuerza automáticamente la recompilación de los ficheros cuyo código fuente haya cambiado después de que el código objeto fuese creado

Limitaciones de “make”

- Basado en un modelo físico de dependencias (no lógico)
- Las especificaciones de dependencias (Makefiles) crecen rápidamente, llegando a ser complejas, difíciles de comprender y “caras” de mantener
- Usa simplemente un modelo de cambio basado en fechas de actualización de ficheros. Puede que cambios en el código fuente no necesiten recompilación
- No permite (fácilmente) especificar la versión de utilización de herramientas como el compilador
- No “finamente” enlazado con herramientas de gestión de configuraciones como RCS

Beneficios GCS

- Reduce el esfuerzo necesario para gestionar y realizar el cambio - mejora la productividad
- Conduce a una mejora de la integridad y seguridad del software - incremento de la calidad
- Genera información sobre el proceso - mejora de la gestión del control
- Mantiene una base de datos de desarrollo de software - mejor registro y seguimiento de informes

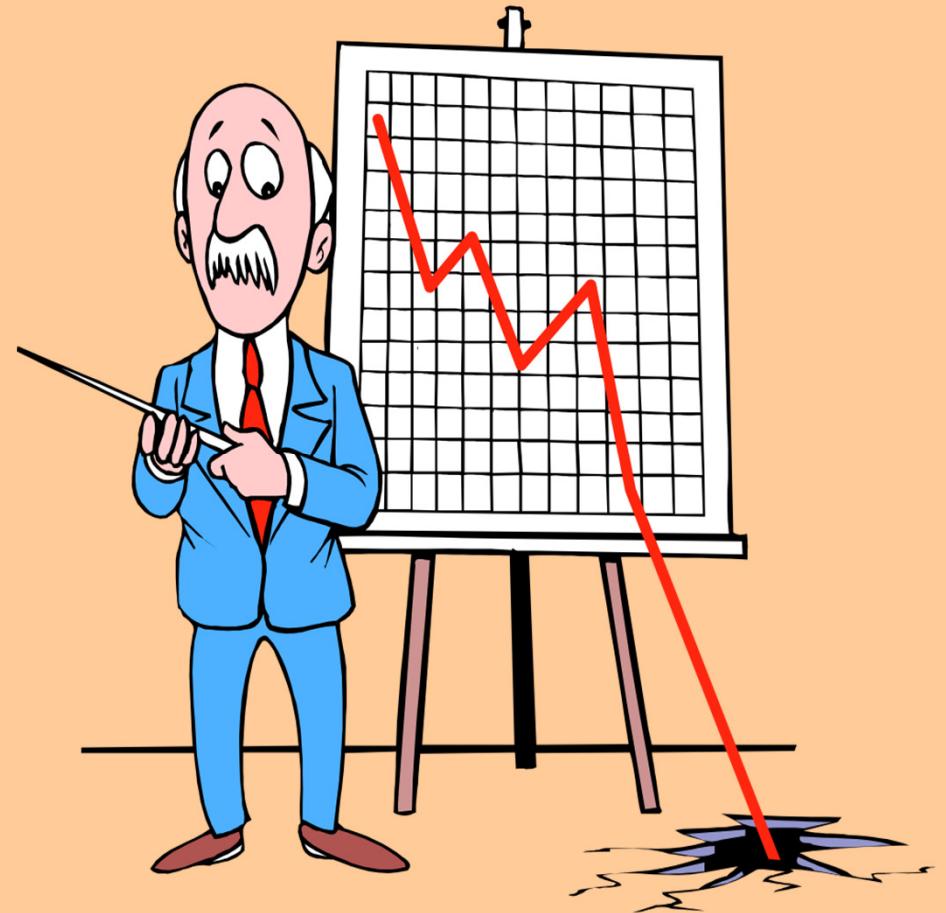
Tema 6.

Monitorización y control.

- Monitorización de una agenda
- Métricas de seguimiento y control: holguras y EVA
- Control de una agenda

Se puede mejorar!

- El 70% de los proyectos:
 - Cuestan más de lo presupuestado
 - Se entregan más tarde de lo planificado
- El 52% de los proyectos:
 - Se entregan con un 189% de lo presupuestado
- Y otros, después de invertir tiempo y dinero, simplemente nunca se entregan



Monitorización de una agenda

- Monitorizar (hacer un seguimiento) una agenda consiste en comprobar si la agenda real se ajusta a la planificada
 - ✿ A la agenda creada inicialmente la denominaremos agenda planificada (muestra nuestra intención inicial, a partir de información planificada)
 - ✿ A medida que el proyecto progresá se creará una agenda real (muestra lo que realmente está ocurriendo, a partir de información real)
- Para monitorizar la agenda necesito hacer uso de diversas **MÉTRICAS** del proyecto
 - ✿ Una métrica es cualquier tipo de medición que proporciona un valor **cuantitativo** para indicar el grado en el que un sistema, componente o proceso posee un determinado atributo
 - ✿ Por ejemplo la holgura libre de una actividad indica cuánto puedo retrasar esa actividad sin afectar a las siguientes ni a la finalización del proyecto

Métricas de seguimiento del proyecto

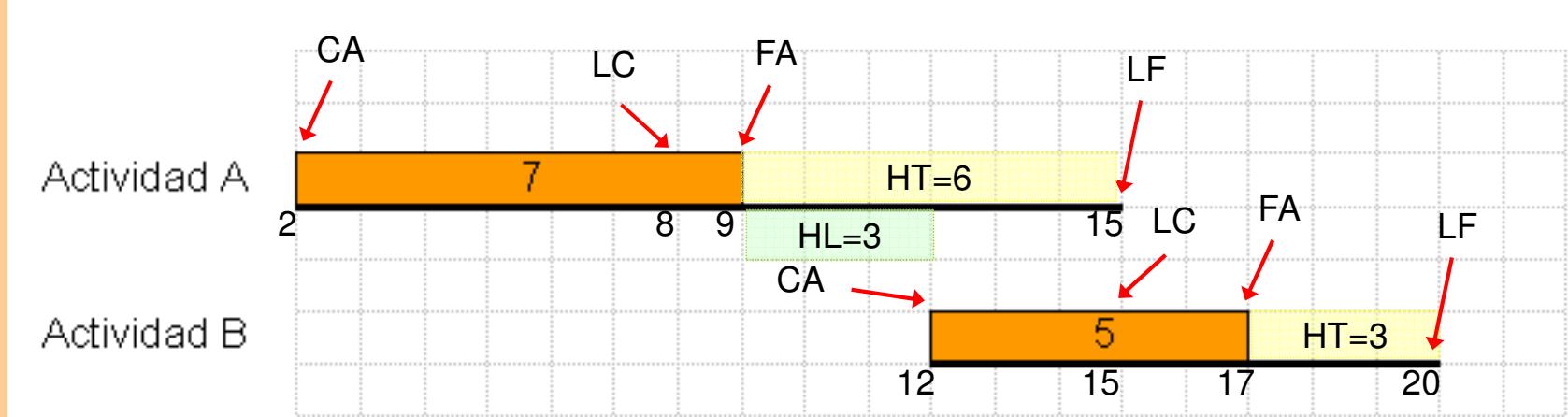
- Fechas de inicio/fin
 - Duraciones
 - Holguras totales y libres
 - Análisis del Valor Acumulado (EVA)
-
- La cuestión es: ¿cómo utilizo las métricas anteriores para MONITORIZAR Y CONTROLAR el proyecto?
 - Recordemos que nuestro objetivo es TOMAR LAS DECISIONES adecuadas que garanticen el éxito del proyecto:
 - ✿ cumplir con los plazos temporales, costes y expectativas del usuario PREVISTOS

Programación de la agenda prevista

- Una vez que tenemos las entradas: **actividades, dependencias, duraciones y recursos**, se realizan los siguientes **CÁLCULOS** (entre otros):
 - ✿ Fechas de inicio y fin más tempranas y tardías de las actividades
 - ✿ Holguras totales y libres de las actividades
 - ★ Tiempo que una actividad puede retrasarse sin retrasar el proyecto (**HOLGURA TOTAL**: HT)
 - ★ Tiempo que una actividad puede retrasarse permitiendo que las actividades siguientes puedan comenzar lo más pronto posible (**HOLGURA LIBRE**: HL)
 - ✿ Camino crítico
 - ★ Secuencia de actividades con holgura total 0
 - ✿ Cálculos de EV (Earned Value)

Para tener nuestra agenda sólo nos faltaría establecer una
FECHA DE INICIO

Cálculo de Holguras: veamos un ejemplo gráfico

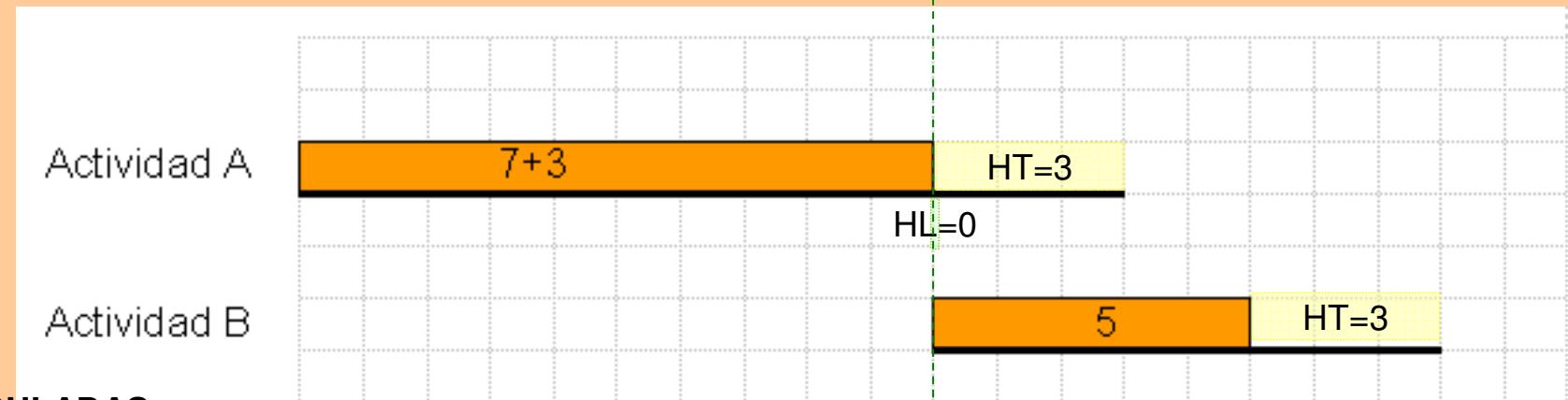


■ $H_T(A) = 15 - 2 - 7 = 6; H_L(A) = 12 - 2 - 7 = 3$

■ $H_T(B) = 20 - 12 - 5 = 3;$

■ ¿Qué pasa si A se retrasa 3 días?

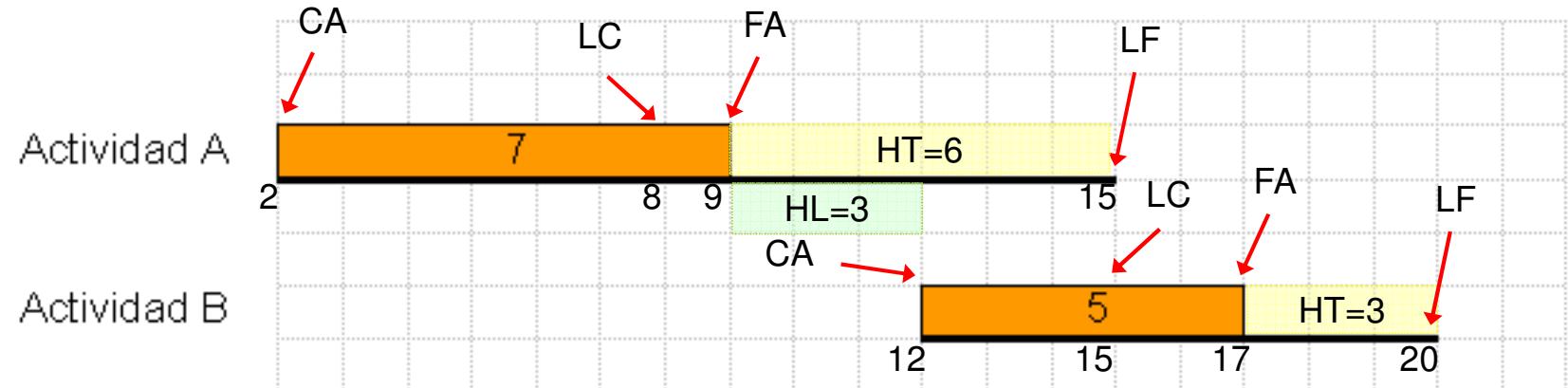
- ✓ El proyecto NO se retrasa
- ✓ B puede comenzar lo antes posible



FECHAS CALCULADAS

CA: Comienzo anticipado (fecha más temprana de comienzo) FA: Fin anticipado (fecha más temprana de fin)
LC: Límite de comienzo (fecha más tardía de comienzo) LF: Límite de fin (fecha más tardía de fin)

Holguras: cómo afectan las actividades siguientes

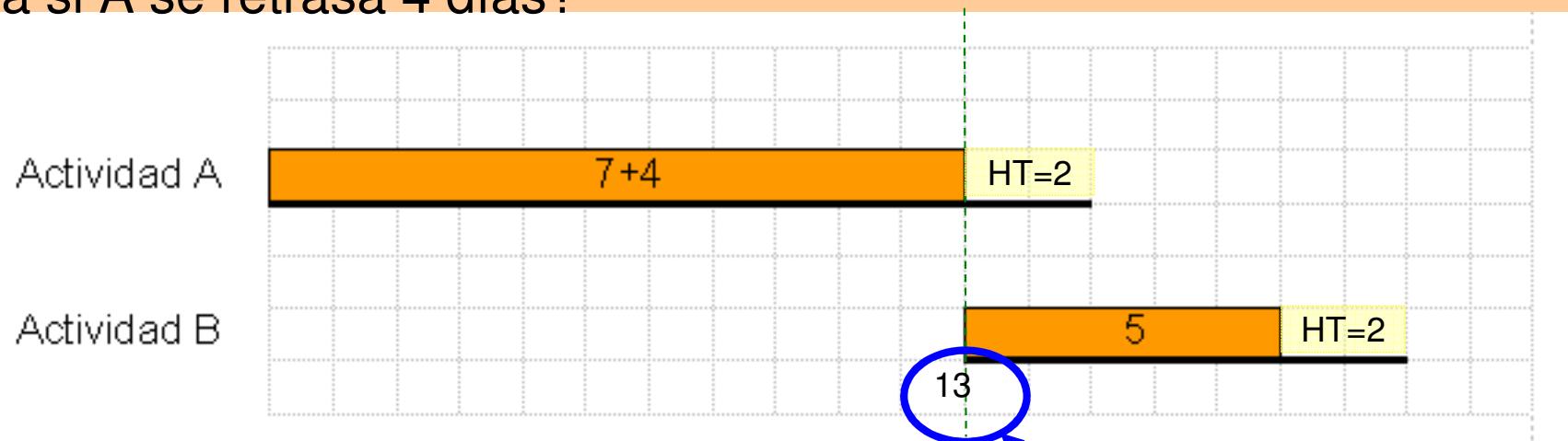


- $H_T(A) = 15 - 2 - 7 = 6; H_L(A) = 12 - 2 - 7 = 3$

- $H_T(B) = 20 - 12 - 5 = 3;$

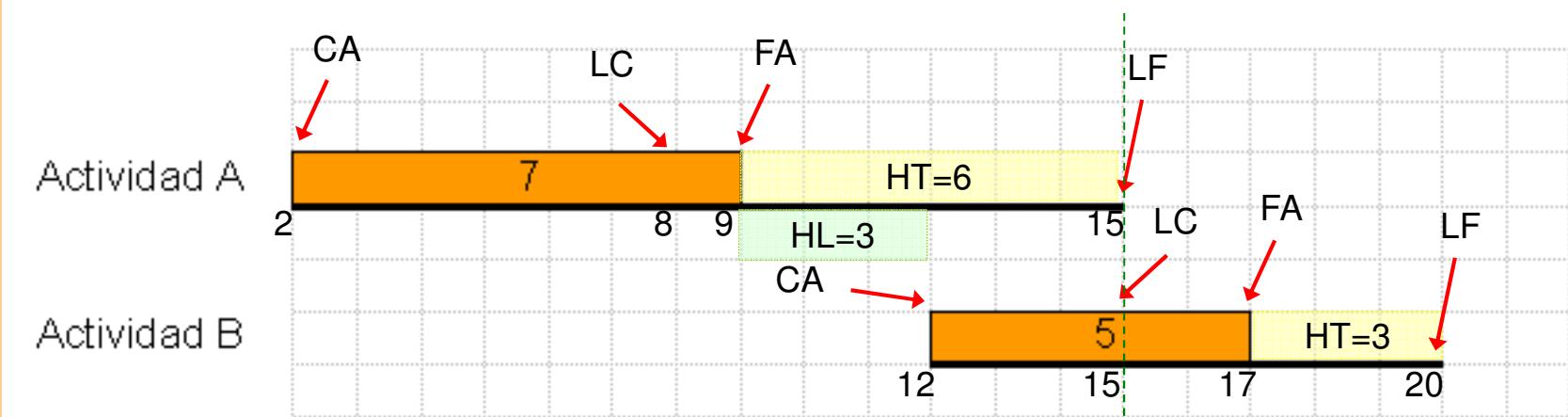
- ¿Qué pasa si A se retrasa 4 días?

✓ El proyecto NO se retrasa
✓ B reduce su HT en 1 unidad



B tiene que empezar 1 día más tarde

Holguras: nuevos caminos críticos

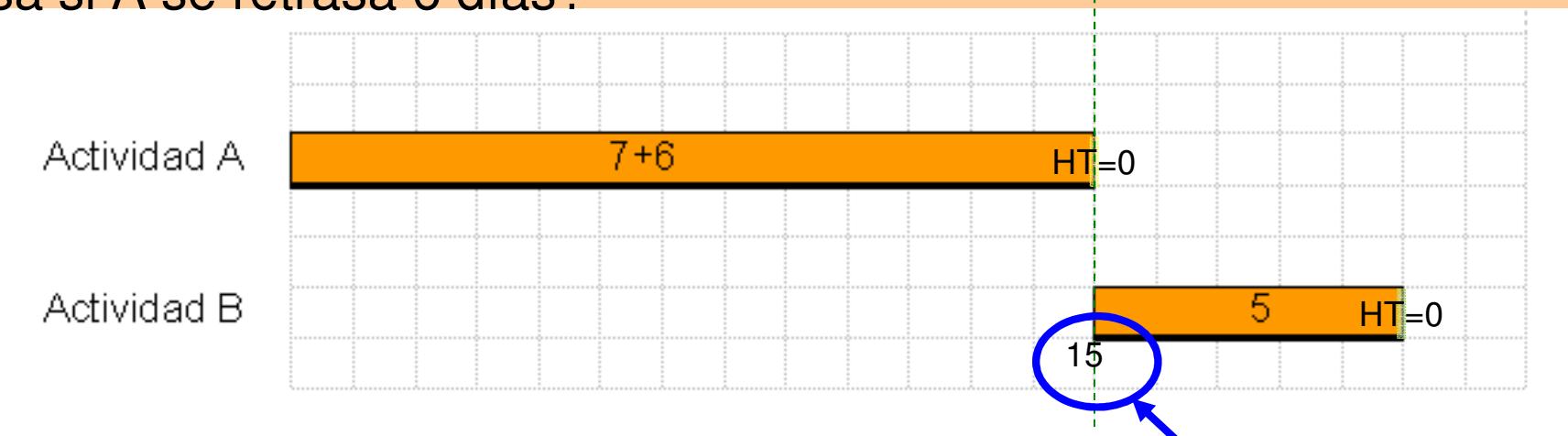


■ $H_T(A) = 15 - 2 - 7 = 6; H_L(A) = 12 - 2 - 7 = 3$

■ $H_T(B) = 20 - 12 - 5 = 3;$

■ ¿Qué pasa si A se retrasa 6 días?

- ✓ El proyecto NO se retrasa
- ✓ B se convierte en CRITICA



Ms Project y Fechas/Holguras programadas

Campos de FECHA:

- Comienzo anticipado, Límite de comienzo
- Fin anticipado, Límite de finalización

	Nombre de tarea	Comienzo anticipado	Límite de comienzo	Fin anticipado	Límite de finalización
1	- Proyecto 1	jue 06/03/08	jue 06/03/08	jue 03/04/08	jue 03/04/08
2	A	jue 06/03/08	vie 07/03/08	lun 17/03/08	mar 18/03/08
3	B	jue 06/03/08	jue 06/03/08	vie 14/03/08	vie 14/03/08
4	C	jue 06/03/08	mar 18/03/08	vie 07/03/08	mié 19/03/08
5	D	mar 18/03/08	mié 19/03/08	mié 19/03/08	jue 20/03/08
6	E	jue 17/03/08	jue 17/03/08	jue 20/03/08	jue 20/03/08

Campos de HOLGURA:

- Demora permisible (Holgura libre)
- Margen de demora total (Holgura total)

	Nombre de tarea	Demora permisible	Margen de demora total
1	- Proyecto 1	0 días	0 días
2	A	0 días	1 día
3	B	0 días	0 días
4	C	5 días	8 días
5	D	1 día	1 día
6	E	0 días	0 días
7	F	0 días	0 días

La vista de Gantt detallado muestra gráficamente las holguras libres

Earned Value Analysis (EVA)

- El EV (*earned value*) o valor acumulado es una métrica que proporciona una información CUANTITATIVA del PROGRESO de un proyecto
 - ✿ A cada tarea se le asigna un valor devengado basado en su porcentaje estimado del valor total
- Permite vislumbrar dificultades en la agenda antes de que éstas puedan ser aparentes.
 - ✿ Esto permite al gestor del proyecto tomar acciones correctivas antes de que el proyecto "entre en crisis"



A menos que realicemos un seguimiento del EV, realmente no tendremos idea de qué está pasando en nuestro proyecto!!!

EVA: Parámetros

- EVA básicamente compara tres tipos de información:
 - ¿Cuánto trabajo (del que se ha planificado) debería haberse completado hasta el momento? (Valor planificado: BCWS)
 - ¿Cuánto se ha gastado hasta el momento? (Coste real: ACWP)
 - ¿Cuál es el valor, en términos del coste de línea base, del trabajo realizado hasta el momento? (Valor acumulado: BCWP)

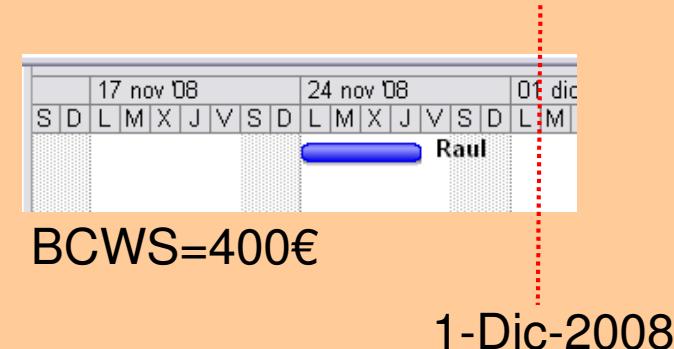
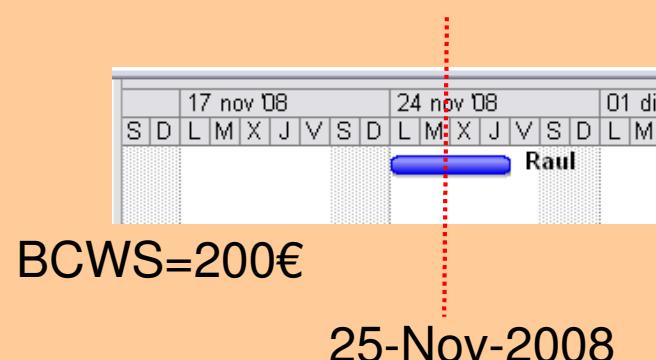


¡IMPORTANTE!

Un análisis de valor acumulado
SIEMPRE se hace tomando como
referencia un instante de tiempo
concreto del desarrollo del proyecto!!!

BCWS- Ejemplo

- Budgeted Cost of Work Scheduled
- Supongamos una tarea X con una duración de 4 días (desde el lun 24-Nov-2008 hasta el jue 27-Nov-2008), con un recurso asignado cuyo salario es 100€/día
 - El coste previsto para la tarea, por lo tanto, es de 400€ (100×4)
 - Si calculamos BCWS el día 25-Nov-2008 tendrá un valor de 200€, puesto que la tarea debería estar medio realizada
 - Si calculamos BCWS el día 1-Dic-2008 tendrá un valor de 400€, puesto que la tarea debería estar completada

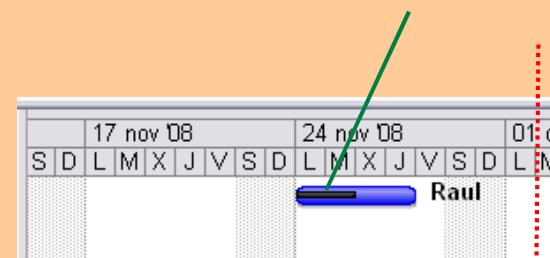


BCWP- Ejemplo

- Budgeted Cost of Work Performed
- Supongamos que la tarea X anterior solamente se ha "completado" hasta la mitad (se ha hecho el 50% de lo que se había pedido):
 - El coste previsto para la tarea sigue siendo de 400€ (100×4)
 - Si calculamos BCWP el día 25-Nov-2008 tendrá un valor de 200€: el 100% del valor planificado
 - Si calculamos BCWP el día 1-Dic-2008 tendrá un valor de 200€: el 50% del valor planificado



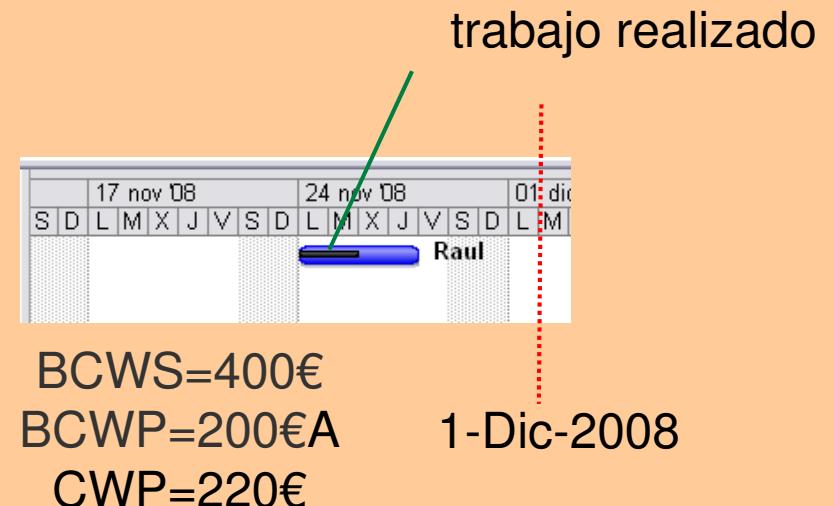
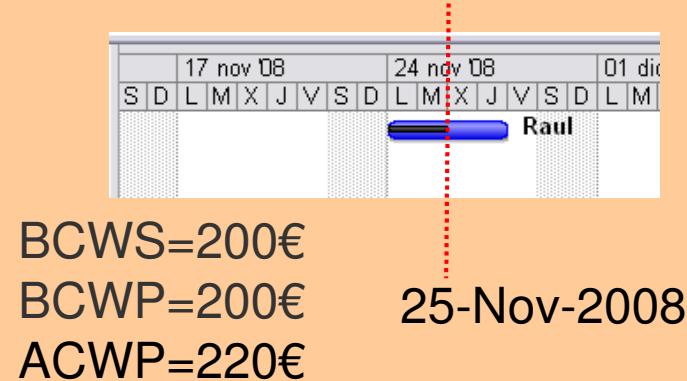
BCWS=200€
BCWP=200€ 25-Nov-2008



BCWS=400€
BCWP=200€ 1-Dic-2008

ACWP- Ejemplo

- Actual Cost of Work Performed
- Supongamos que el recurso asignado a la tarea X realmente recibe 110€/dia para esta tarea (en lugar de los 100€ inicialmente previstos):
 - El coste previsto para la tarea sigue siendo de 400€ (100×4)
 - Si calculamos ACWP el día 25-Nov-2008 tendrá un valor de 220€ (110×2)
 - Si calculamos ACWP el día 1-Dic-2008 tendrá un valor de 220€ (110×2)



¿Cómo funciona EVA?

- Indicadores de **PROGRESO**:
 - Schedule Variance (SV) = BCWP - BCWS
 - Schedule Performance Index (SPI) = BCWP / BCWS
 - Si BCWP > BCWS la tarea/proyecto va adelantada según la agenda planificada
 - P.ej. un SPI de 1,5 significa que sólo ha utilizado el 67% del tiempo planeado para completar una parte de la tarea en un determinado periodo de tiempo (BCWS = 0,67 BCWP)
- Indicadores de **PRODUCTIVIDAD**
 - Cost Variance (CV) = BCWP - ACWP
 - Cost Performance Index (CPI) = BCWP / ACWP
 - Si BCWP > ACWP la tarea/proyecto está gastando menos de lo planificado
 - P.ej. un CPI de 0,8 significa que se está gastando un 25% más de lo que estaba planificado (por cada euro presupuestado se está gastando 1,25€) (ACWP = 1.25 BCWP)
- Si tenemos una buena productividad, y un progreso lento: NOS FALTA GENTE!!!

Ejemplo EVA: BAC y BCWS

- BAC: Budget at Completion (Cantidad de trabajo planificado al final del proyecto/tarea)

	Task Name	BAC	BCWS
0	Manual Project	19,400.00	14,530.00
1	+ Weekly meetings	0.00	0.00
10	Content	10,450.00	8,900.00
11	Design structure	2,750.00	2,750.00
12	Write body text	6,150.00	6,150.00
13	Set page layouts	1,550.00	0.00
14	Exercises	8,950.00	5,630.00
15	Create exercises	5,300.00	5,300.00
16	Test exercises	1,650.00	330.00
17	Create contents & index	2,000.00	0.00
18	Manual completed	0.00	0.00

Se debería haber realizado hasta fecha un 75% del trabajo planificado del proyecto

Tarea que debería haberse completado (BCWS=BAC)

Todavía no ha terminado (BCWS < BAC)

Tarea que no ha comenzado (BCWS=0)

EVA: BCWS y BCWP

BCWP: Budget Cost of Work Performed (Earned Value!!!!)

- BCWP = Baseline Cost * Percent Complete

	Task Name	BCWS	BCWP
0	Manual Project	14,530.00	13,700.96
1	Weekly meetings	0.00	0.00
10	Content	8,900.00	6,750.96
11	Design structure	2,750.00	2,750.00
12	Write body text	6,150.00	4,000.96
13	Set page layouts	0.00	0.00
14	Exercises	5,630.00	6,950.00
15	Create exercises	5,300.00	5,300.00
16	Test exercises	330.00	1,650.00
17	Create contents & index	0.00	0.00
18	Manual completed	0.00	0.00

El proyecto lleva un **retraso** de trabajo de 829.04 (SV=-829.04), ha utilizado un 6,4% más del tiempo planificado (SPI=0,94)

Tarea con un **retraso** de 2149.04 sobre lo planificado hasta la fecha, ha consumido un 54% más del trabajo planificado (SPI =0,65)

Tarea con un **adelanto** de 1320 sobre lo planificado hasta la fecha, sólo ha utilizado el 20% del tiempo planificado para realizar la parte de tarea planificada (SPI =5)

- SI BCWP > BCWS: La tarea/proyecto va ADELANTADA según la agenda
- SI BCWP < BCWS: La tarea/proyecto va con RETRASO según la agenda

EVA: BCWP y ACWP

ACWP: Actual Cost of Work Performed

	Task Name	BCWP	ACWP
0	Manual Project	13,700.96	14,037.50
1	Weekly meetings	0.00	0.00
10	Content	6,750.96	8,047.50
11	Design structure	2,750.00	2,750.00
12	Write body text	4,000.96	5,297.50
13	Set page layouts	0.00	0.00
14	Exercises	6,950.00	5,990.00
15	Create exercises	5,300.00	4,660.00
16	Test exercises	1,650.00	1,330.00
17	Create contents & index	0.00	0.00
18	Manual completed	0.00	0.00

El proyecto va **por encima** de lo presupuestado ($CV=-336.54$), ha gastado un 2,4% más del presupuesto planificado ($CPI=0,98$)

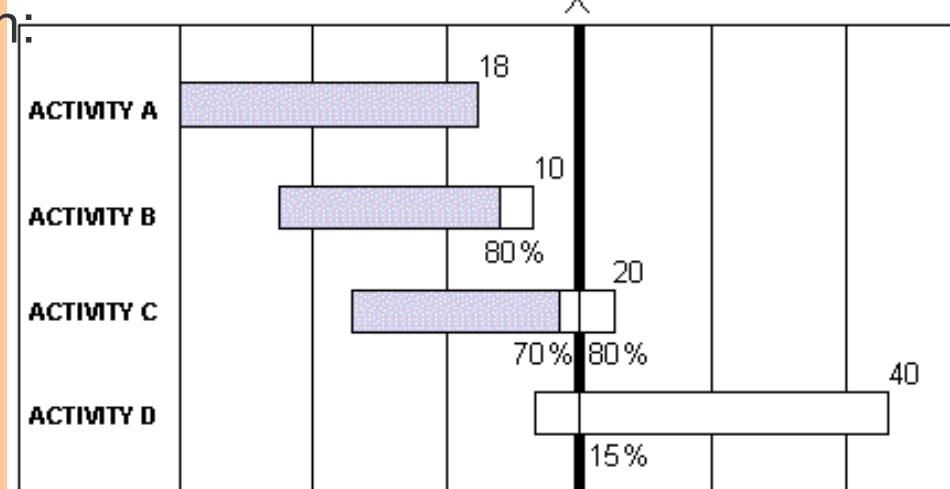
La tarea va **por encima** de lo presupuestado en 1296,54€, por cada euro presupuestado se ha gastado 1,33€ ($CPI =0,75$)

La tarea va **por debajo** de lo presupuestado en 320€, por cada euro presupuestado sólo se ha gastado 0,80€ ($CPI =1,24$)

- SI BCWP > ACWS: La tarea/proyecto está por DEBAJO de lo presupuestado
- SI BCWP < ACWS: La tarea/proyecto está por ENCIMA de lo presupuestado

EVA: Ejemplo

- Determina BCWS, BCWP y realiza un análisis EVA, a partir de la información de la siguiente figura y suponiendo que ACWP tiene un valor de 45\$ información:



- Valor planificado (BCWS) = $18 + 10 + 16 + 6 = \$50$

- Valor earned (BCWP) = $18 + 8 + 14 + 0 = \$40$

- Coste actual (ACWP) = 45\$

- SV = $40 - 50 = -\$10$

- SPI = $40 / 50 = 0.8$

- CV = $40 - 45 = -\$5$

- CPI = $40 / 45 = 0.89$

El proyecto necesita un 25% más del tiempo planificado para completar el proyecto

El proyecto está gastando $\approx \$1.13$ por cada \$1.00 de trabajo presupuestado que se completa

Ejercicio propuesto

- Realiza un análisis cuantitativo del progreso de las siguientes tareas:

BAC (CPF)	BCWS (CPTP)	BCWP (CPTR)	ACWP (CRTR)	SV (VP)	CV (VC)	SPI (IRP)	CPI (IRC)	Tareas
1000	1000	1000	1200					T1
1500	1500	1000	1500					T2
6000	5000	4000	4500					T3
1000	500	0	0					T4
1000	500	500	750					T5
2000	1000	1500	1500					T6

- Suponiendo que nuestro proyecto está formado exclusivamente por las tareas de la tabla anterior, ¿podríamos deducir que falta gente en el proyecto?

Ejercicio propuesto: guía para la solución

- A continuación mostramos el análisis para la Tarea T1:

BAC (CPF)	BCWS (CPTP)	BCWP (CPTR)	ACWP (CRTR)	SV (VP)	CV (VC)	SPI (IRP)	CPI (IRC)	Tareas
1000	1000	1000	1200	0	-200	1	0,8333	T1

La Tarea (1) debería haber terminado (ya que BAC = BCWS).

La tarea se está ejecutando según la agenda (ya que SV=0), de hecho ya ha terminado. Hemos gastado 200 de más (ya que CV es -200)

Pero hemos gastado 1,20 euros por cada 1 euro de coste presupuestado ($1/0,83=1,20$, o también $1200/1000$).

Ms Project y EVA

- BCWS = CPTP (Costo Presupuestado del Trabajo Programado)
- ACWP = CRTR (Costo Real del Trabajo Realizado)
- BCWP = CPTR (Costo Presupuestado del Trabajo Realizado)
- Tablas:
 - Valor acumulado
 - Indicadores de costo del valor acumulado

Datos de monitorización y control en Ms Project

- Con Ms Project, se utilizan cinco tipos de información para poder analizar el progreso al realizar el seguimiento de las tareas de un proyecto:
 - duración, trabajo, fecha de comienzo, fecha de fin, y costo
- Los cambios en cada uno de estos campos permiten evaluar el progreso:
 - **Planificado:** información programada de los campos anteriores. Un plan de Línea base es el plan original que se guarda y se utiliza para monitorizar y controlar el progreso
 - **Programado:** información actual más actualizada de los campos anteriores (duración, trabajo, fechas de comienzo y fin, y costos PROGRAMADOS)
 - **Real:** información de lo que ha ocurrido realmente de los campos anteriores (duración, trabajo, fechas de comienzo y fin, y costos REALES)
 - **Restante:** información programada - información real (de trabajo, costo y duración)

Ms Project y seguimiento agenda

- Para hacer un seguimiento tenemos que:
 - ✿ Guardar una **LÍNEA BASE** del proyecto
 - ★ Proyecto→Herramientas→Establecer línea de base
 - ★ Los campos: duración, trabajo, comienzo, fin, costo se guardan como duración, trabajo,..., costo PREVISTOS
 - ✿ Establecer una **FECHA DE ESTADO**
 - ★ Proyecto→Información del proyecto →Fecha de estado
 - ✿ Introducir la **INFORMACIÓN REAL** del proyecto
 - ★ Herramientas→Seguimiento→Actualizar tareas
 - ★ Campos: duración, trabajo, comienzo, fin, costo REALES
 - ✿ Comparar el **PROGRESO** con una vista "Gantt de Seguimiento"
 - ★ Compara la programación de la línea base con la programación real

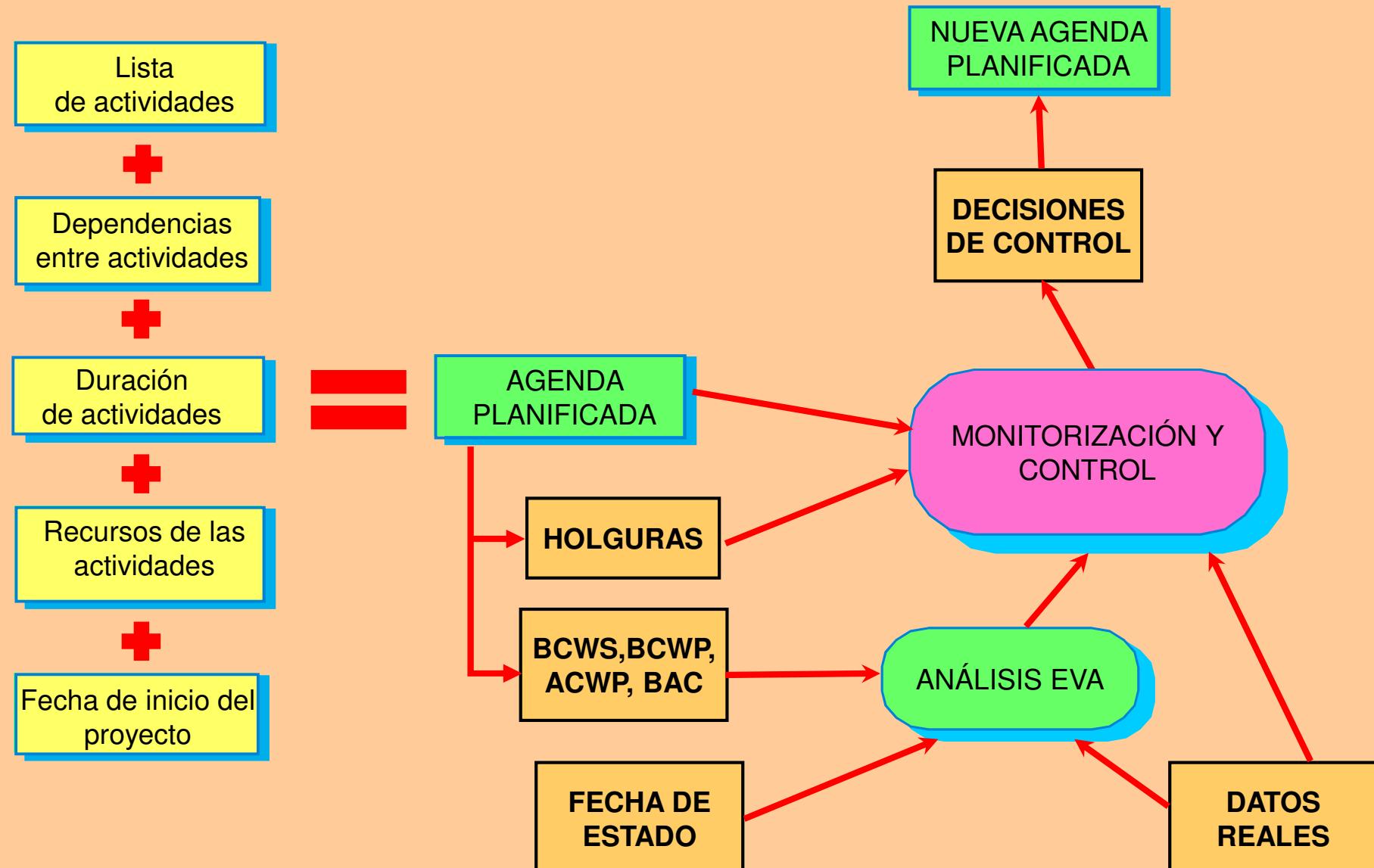
Control de una agenda: holguras

- Controlar una agenda consiste en realizar las acciones necesarias para que las discrepancias entre la agenda real y la planificada sean mínimas
- Las actividades críticas NO se pueden retrasar
- Las actividades no críticas pueden retrasarse siempre y cuando el retraso no supere su holgura total
 - ✿ Si una actividad consume toda (o parte) de su holgura total, puede afectar a la holgura total de la(s) actividad(es) siguiente(s)
 - ✿ La(s) actividad(es) siguiente(s) puede(n) convertirse en crítica(s)
 - ✿ Por ejemplo: A precede B
 - ★ $HT(A) = 3$ y $HL(A)=2$
 - ★ $HT(B) = 1$
 - ★ Si A se retrasa 3 unidades, B se convierte en crítica

Control de una agenda: EVA

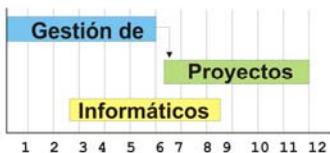
- Como hemos visto, los indicadores de valor acumulado: variaciones (SV, CV) o ratios (SPI, CPI):
 - ✿ pueden ayudar a determinar si queda suficiente dinero en el presupuesto y si el proyecto acabará a tiempo
- Las variaciones:
 - ✿ positivas: pueden permitir reasignar dinero y recursos de tareas con variaciones positivas a otros con variaciones negativas
 - ✿ negativas: pueden indicar que quizá convenga aumentar el presupuesto o aceptar márgenes de beneficio reducidos
- Los ratios:
 - ✿ SPI es un indicador de progreso; CPI es un indicador de productividad
 - ✿ Si $CPI > 1$ y $SPI < 1$ necesitamos contratar a más gente
 - ✿ Si $CPI < 1$ puede que estemos haciendo trabajo no planificado, o que hayamos estimado mal

RESUMEN



Referencias

- PERT: holguras
 - <http://www.adeak.com/2010/01/what-is-float-or-slack-in-a-project-schedule/>
- Earned Value Analysis
 - <http://www.projectlearning.net/pdf/I2.1.pdf>
 - http://www.hyperhot.com/pm_cscs.htm



GESTIÓN DE PROYECTOS INFORMATICOS

Convocatoria de JUNIO. TEORIA

9 de junio de 2016

Apellidos, Nombre: _____

Grupo de prácticas: _____

- Indica 5 factores que afecten a la estimación de costes de un proyecto. Propón una medida para ayudar a minimizar el impacto de cada uno de estos factores. (2,5p)

Oportunidad de mercado: A través de calidad y precio. Para evitar que otras compañías entren en mi mercado diferencia mi producto de la competencia. Haría promociones para que a esos nuevos desarrolladores les cueste más competir y entrar.

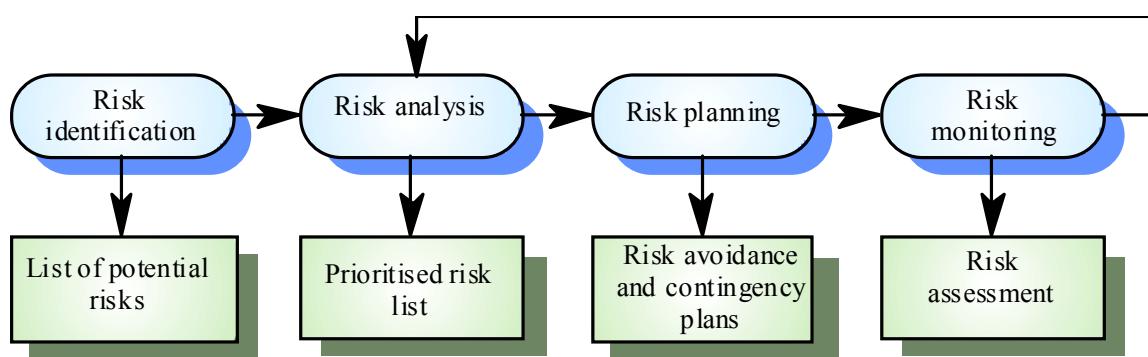
Incertidumbre en la estimación de costes: La parte clave es guardar un histórico de las estimaciones y llevar una contabilización de los costes reales de cada proyecto. En segundo lugar tendría en cuenta márgenes de error en las estimaciones. Luego propondría medidas con el fin de ir estudiando en cada proyecto los errores cometidos, respecto a las estimaciones, tratando de que estos fuesen menores.

Términos contractuales: Para minimizar los costes intentaría que la venta fuese a través de licencias, manteniendo nosotros la propiedad del software y de sus derechos comerciales.

Volatibilidad de los requerimientos: Lo que haría es mantener una comunicación con el cliente a lo largo del proyecto. Le haría validar los requerimientos, utilizaría un prototipo para comprobar que hemos entendido bien el problema, etc.

Salud financiera: Para minimizar el impacto cobraría por adelantado parte del proyecto, crearía un fondo para autofinanciar futuros proyectos, estudiaría fórmulas de financiación, pediría subvenciones, etc.

- Crea un plan de riesgos para un proyecto ficticio (identifica sólo 3 riesgos). Indica los pasos seguidos y crea los documentos oportunos. Detalla como harías el seguimiento y control de dicho proyecto atendiendo a los riesgos identificados. (2,5p).



:

Identificar riesgos:

- a. Es imposible seleccionar personal con las habilidades requeridas para el proyecto.
- b. Baja en el personal.
- c. Los problemas financieros en la organización causan reducciones en el presupuesto del proyecto.

Analizar riesgos:

- a. Probabilidad alta. Efecto catastrófico.
- b. Probabilidad media. Efecto serio.
- c. Probabilidad baja. Efecto catastrófico.

Plan de prevención y minimización.

- d. Hacer una campaña de selección donde se difundan muchos los puestos ofertados dentro de personal especializado. Prever cursos de formación.
- e. Utilizar una estructura organizativa democrática descentralizada, donde todos hagamos todo tipo de tareas.
- f. Preparar un informe para justificar la importancia de nuestro proyecto.

Seguimiento:

- a. Revisar número de candidatos en otras convocatorias. Trabajar conjuntamente con las organizaciones educativas con el fin de conocer los perfiles.
- b. Llevar un control de ausencias. Ver el ánimo del grupo. Hacer actos sociales con mi personal para conocer su situación.
- c. Mantener contacto con directivos de la empresa, gestores de otros proyectos de nuestra empresa y mercado con el fin de ir conociendo el estado del sector.

3. ¿Qué relación guarda un modelo de proceso con la agenda de un proyecto y con la estimación de costes? Pon un ejemplo que justifique la respuesta. (2,5p)

El modelo de proceso condiciona qué actividades y en qué orden se incluirán en la agenda del proyecto, necesaria para poder conocer el tiempo y el coste total de desarrollo, y que son responsabilidad del gestor de un proyecto.

El modelo de proceso facilita la subdivisión de actividades en otras más "pequeñas", ya que proporciona el conjunto de actividades que hay que realizar. Estas actividades pueden subdividirse y estructurarse convenientemente, por ejemplo utilizando representaciones WBS que son muy útiles para estimar los costes, en donde, además, se favorece el uso de una estimación ascendente o descendente.

Las estadísticas de proyectos anteriores nos van a ayudar en las estimaciones, más aún cuando el proyecto se haya desarrollado con el mismo modelo de proceso. En el caso de UP incluso nos podría decir los % de las estimaciones que se usan en cada fase.

Si se sigue un modelo de proceso UP nos dará lugar a las 4 fases (inicio, elaboración, construcción y transición), así como a un conjunto de n iteraciones y a una serie de tareas de cualquier disciplina dentro de las iteraciones. Por otro lado las relaciones de precedencia de dichas actividades se corresponden con la secuencia temporal de las actividades que se llevan a cabo en el modelo UP.

4. Dados los siguientes datos de un proyecto. Explica cuál es la situación del proyecto para cada una de las tareas. ¿La tarea 1 se ha terminado? ¿Cómo podemos solucionar los problemas del proyecto? (2,5p)

Tarea	BCWS	BCWP	ACWP
Tarea R	2800	1800	2600
Tarea 1	600	600	1000
Tarea 2	1200	0	0
Tarea 3	1000	1200	800
Tarea 4	0	0	800

La situación del proyecto es que va retrasado y su productividad es baja (gastando de más).

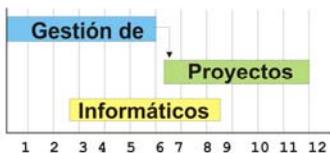
La tarea 1 va bien respecto a la agenda, pero en ella se ha gastado más de lo previsto. No podemos saber si se ha terminado, para ello nos haría falta conocer su BAC.

La tarea 2 debería de haber comenzado, pero todavía no ha hecho.

La tarea 3 ha comenzado, pero va retrasada, sin embargo su productividad es buena (se está gastando menos de lo previsto).

La tarea 4 no debería haber comenzado y no lo ha hecho. El valor ACWP es erróneo, sino ha empezado a realizarse no debería de tener imputados gastos.

Para resolver los problemas del proyecto podríamos asignar recursos de la tarea 3 a 2, ya que la 3 va adelantada y la 2 retrasada. También podríamos contratar a más gente para la 2 ya que en la 3 estamos gastando menos de lo esperado.



GESTIÓN DE PROYECTOS INFORMATICOS

Convocatoria de JUNIO. PRÁCTICAS

9 de junio de 2016

Apellidos, Nombre: _____

Grupo de prácticas: _____

1.- Dentro de un plan de proyecto planificado en Microsoft Project. ¿Cómo podemos definir los objetivos (metas)? (0,5p)

Los podemos definir con tareas e duración 0 (hitos). Otra alternativa es en el propio título de la fase o iteración. Y la última definiendo una nota.

¿Se podrían fijar los objetivos de las fases y las iteraciones en la planificación inicial? Pon los objetivos de las fases definidos a lo largo de tus prácticas de la asignatura (0,5p)

Fase de inicio. Estimaciones y plan inicial. Estudio de viabilidad del proyecto.

Fase de elaboración. Prototipo de la interfaz gráfica de la aplicación (formularios)

Fase de Construcción. Implementar la capa de lógica de negocio y acceso a datos.

Fase de transición. Pruebas del sistema. Creación de soportes de distribución y manuales.

2.- Dentro de la LPOD, que son los códigos tipo? Nombra 3 códigos tipo. (0,5p)

Los códigos tipo son formularios NOTA prerellenados y simplificados para poder dar de alta un fichero de datos dentro de la agencia de protección de datos de una manera muy sencilla. Así mismo, permite notificar de forma simplificada, los ficheros de titularidad privada de comunidades de propietarios, clientes, libro recetario de las oficinas de farmacia, pacientes, gestión escolar, nóminas, recursos humanos y videovigilancia y los de titularidad pública de recursos humanos, gestión del padrón, gestión económica o control de acceso.

A la hora de registrar software, ¿Qué dos alternativas tenemos? ¿Cuál se suele utilizar? ¿Para esta alternativa a través de quién debemos de hacerlo y que impresos debemos de rellenar? (0,5p)

Tenemos la opción de hacerlo a través de patente de software y de registro de la propiedad intelectual. El más común (99%) es a través del registro de propiedad intelectual. Esto se hace a través del ministerio de Educación, cultura y Deporte. Hay que llenar los impresos de autores o impreso de otros titulares originarios, y el modelo de programas de ordenador.



GESTIÓN DE PROYECTOS INFORMATICOS

Convocatoria de JULIO. TEORIA

6 de julio de 2016

Apellidos, Nombre: _____

Grupo de prácticas: _____

- 1.- Explica qué es la visibilidad de un proceso software y por qué es importante que un proceso sea visible. Pon tres ejemplos que den visibilidad a un proyecto (2.5 p)

La visibilidad de un proceso software consiste en conocer en cada momento exactamente qué es lo que se está haciendo en cada momento del desarrollo. Dado que el software es un producto intangible, la única forma de hacer que el proceso de desarrollo muestre el estado del producto en construcción es mediante la documentación generada durante el proceso.

La posibilidad de "ver" en qué estado del desarrollo se encuentra un producto software resulta de vital importancia a la hora de gestionar un proyecto software, de forma que se pueda hacer una planificación efectiva, y monitorizar y controlar el desarrollo de dicho software.

Ejemplos concretos pueden ser: utilizar 3 niveles en el plan, utilización de estándares, hitos (objetivos definidos), Gantt de seguimiento, informe de iteración, etc.

- 2.- Explica en que consiste la estimación con puntos de función. Indica las ventajas de la estimación de puntos de función respecto a líneas de código y de la estimación mediante puntos objeto respecto a puntos de función (2.5 p).

La técnica de estimación de puntos de función se basa en la contabilización de unos contadores (características del programa):

entradas y salidas externas.

interacciones de usuario.

interfaces externas.

ficheros usados por el sistema.

Se asocia un peso con cada uno de estos contadores y los puntos de función se calculan multiplicando cada factor por su peso y sumando todos ellos

Puntos de función frente a líneas de código.

Son independientes del lenguaje de programación

Pueden calcularse a partir de la especificación

Usa información del dominio del problema

Resulta más fácil a la hora de reusar componentes

Se encamina a aproximaciones orientadas a objetos

Puntos objeto frente a puntos de función.

- Son más fáciles de estimar a partir de una especificación que los puntos de función, ya que solamente consideran pantallas, informes y módulos 3GL

- Por lo tanto pueden estimarse en fases tempranas del proceso de desarrollo. En estas etapas resulta muy difícil estimar el número de líneas de código de un sistema.

3.- ¿Qué es el modelo P-CMM? ¿Qué etapas tiene? Explica cada una de ellas. ¿Cuáles son sus objetivos (2,5p)

Pretende ser un marco para la gestión del trabajo realizado por la gente implicada en el desarrollo del software

Es un modelo de cinco etapas

Inicial. Gestión de rec. humanos "ad-hoc"

Repetible. Se desarrollan políticas para mejora de las capacidades (aptitudes)

Definido. Gestión de rec. humanos estandarizada para la organización

Gestionado. Se establecen metas cuantitativas para la gestión de recursos humanos

Optimizado. Se realiza un esfuerzo continuado para mejorar la competencia y motivación en el trabajo

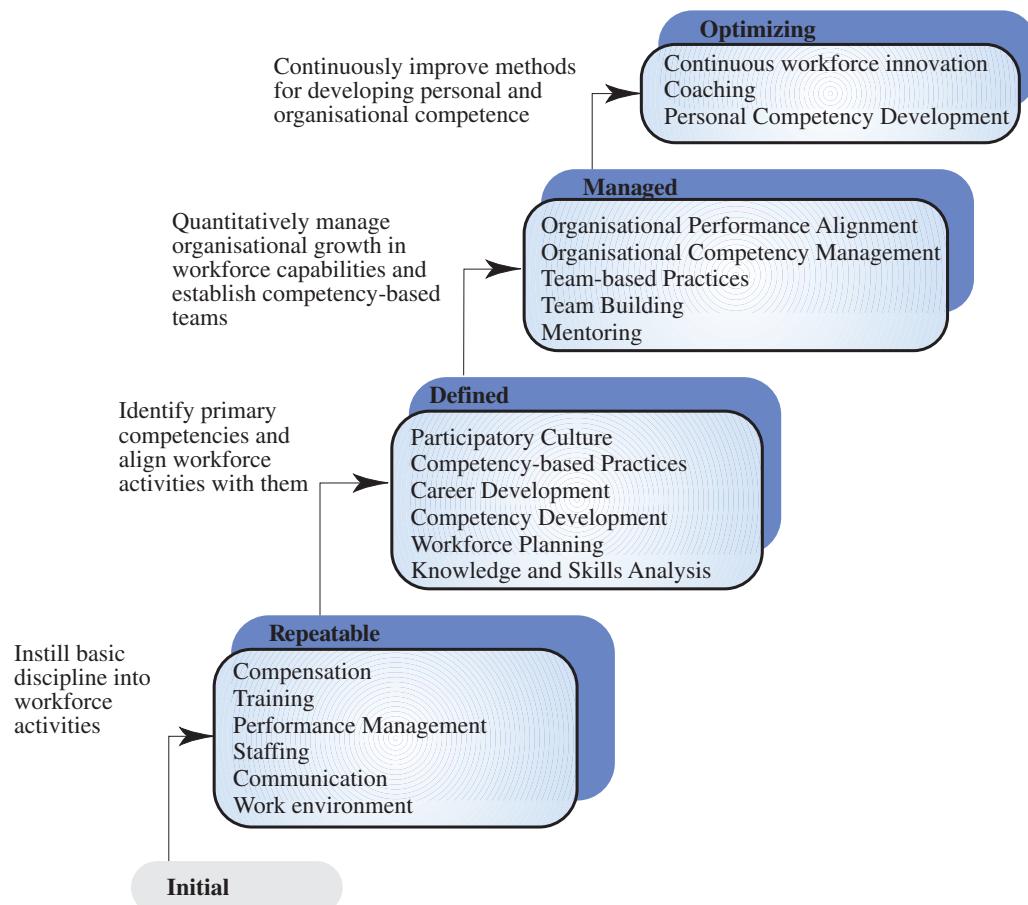
Sus objetivos son:

Mejorar las capacidades de la organización mejorando las capacidades de trabajo de la gente implicada

Asegurar que las capacidades para el desarrollo del software no conciernen a un número pequeño de individuos

Igualar la motivación de los individuos con la de la organización

Ayudar a la "retención" de gente con conocimientos y habilidades críticas



4.- ¿Qué significa que en una fecha determinada una actividad de un proyecto tiene un SPI de 1,5? (0,5 p)

Vamos muy bien respecto al tiempo en el plan. Necesitamos el 50% menos de tiempo para completar el proyecto.

- ¿Qué significa que en una fecha determinada una actividad de un proyecto tiene un CPI de 0,8? (0,5 p)

Que tenemos una productividad baja. Estamos gastando un 25% más de lo que toca.

- Estamos realizando un análisis EVA de un proyecto y resulta que en una fecha determinada el indicador SPI es menor que 1 y el indicador CPI es mayor que 1. ¿Qué significan estos valores? ¿Qué debemos de hacer?

(0,75 p)

Que tenemos una productividad alta, pero vamos retrasados. Debemos de contratar a más gente.

- Estamos realizando un análisis EVA de un proyecto y en una fecha determinada tenemos los siguientes valores para una actividad:

BAC	BCWP	BCWS	ACWP
2000 €	1500 €	1000 €	1500 €

En base a estos datos, ¿Qué podemos afirmar respecto al progreso y coste de esta actividad? Debes justificar tu respuesta realizando los cálculos que consideres necesarios. (0,75 p)

Podemos afirmar que vamos adelantados(BCWP) y gastando justamente lo presupuestado (ACWP). En concreto hemos hecho un 50% más de lo esperado, llevamos hecho un 75% de la tarea cuando planificamos llevar el 50%



GESTIÓN DE PROYECTOS INFORMATICOS

Convocatoria de JULIO. PRÁCTICAS

6 de julio de 2016

Apellidos, Nombre: _____

Grupo de prácticas: _____

1.- Explica los pasos seguidos en prácticas dentro Microsoft Project para asignar recursos. (0,5p). Explica las diferentes formas que existen en Microsoft Project para hacer el seguimiento de un proyecto. (0,5p)

- a) Crearíamos los recursos.
- b) Asignaríamos los recursos a cada una de las tareas usando el filtro de recursos para facilitar la tarea.
- c) Comprobaríamos el uso de recursos mediante el informe de “uso de recursos” con el fin de verificar que no existen sobreasignaciones.

Para hacer el seguimiento de un proyecto en Microsoft Project tenemos las siguientes posibilidades:

- a) Hacer uso de los tiempos early, last y las holguras libres y totales, añadiendo las columnas a la rejilla de datos.
- b) Ver el diagrama Pert.
- c) Visualizando los valores EVA.
- d) Mostrando el Gant de seguimiento.

2.- A la hora de registrar software, ¿Qué diferencia hay entre los impresos de autores1 A-T, Autores TIV y Autores TMC? Pon un ejemplo de uso relacionado con software donde se utilice el impreso A-T, y otro donde se utilice el TIV. (0,5p)

- Autores 1 A-T: a cumplimentar por los titulares de derechos de propiedad intelectual que sean autores y titulares de una obra.

- Autores 1 TIV: a cumplimentar por los titulares que hubieran adquirido los derechos por transmisión inter vivos, ya sea por contrato de cesión o por relación laboral, y siempre que se trate de una primera inscripción de derechos.

- Autores 1 TMC: a cumplimentar por los herederos que hubieran adquirido los derechos por transmisión mortis causa del autor, siempre que se trate de una primera inscripción de derechos.

Ejemplos: Un profesional autónomo crea un programa software y para registrararlo utiliza el impreso A-T, mientras que una empresa de desarrollo con empleados utilizaría el formulario TIV.

¿En el caso de informática, entre que 3 tipos de impresos (autores 2) tenemos? ¿En nuestra práctica cual se eligió? (0.5 p)

- Programas de ordenador.
- Base de datos.
- Páginas Web.

En nuestra práctica se eligió el primero de ellos, programas de ordenador.



GESTIÓN DE PROYECTOS INFORMATICOS

Convocatoria de JULIO

6 de julio de 2015

Apellidos, Nombre: _____

Grupo de prácticas: _____

1. Explica las diferencias entre un presupuesto y un documento de estimación de costes. ¿Qué información debe de contener cada uno de ellos? (2 p)

2.- ¿Los objetivos de la fase de construcción de un proyecto de que dependen? En un proyecto basado en UP y prototipado ¿Cuáles podrían ser los objetivos de la fase de construcción? Utilizando iteraciones de duración media ¿Cuántas iteraciones podríamos planificar detalladamente de la fase de construcción en el primer trimestre del proyecto? ¿Por qué? (2 p)

3.- Indica las tareas que podemos realizar dentro de Microsoft Project ¿En el caso de un retraso en una actividad ¿Qué pasos deberíamos de seguir para intentar minimizar el impacto en el proyecto? (2 p)

4.- Explica qué es la pirámide de necesidades. ¿Por qué es importante conocerla como gestores de proyectos? Pon un par de ejemplos de su uso. (2 p)

5.- Explica la diferencia entre mantenimiento y gestión de configuraciones. Explica la diferencia entre un gestor de versiones (como CVS, subversion, etc) y una base de datos de configuraciones. (2 p)

SOLUCIONES

1. Explica las diferencias entre un presupuesto y un documento de estimación de costes. ¿Qué información debe de contener cada uno de ellos? (2 p)

Las estimaciones de coste son un apartado del plan del proyecto, y es una información interna de la empresa. Indica el esfuerzo necesario para realizar un proyecto (personas/mes), el coste temporal y económico del proyecto.

El presupuesto es un documento que se va a entregar al cliente para informarle sobre los precios de venta, tiempos de entrega y funcionalidades de la aplicación.

La diferencia fundamental es que la estimación habla de costes, mientras que el presupuesto habla en términos de precios de venta (coste+margin error+beneficio) y fechas de entrega (entrega estimada+margin de error).

El documento de estimación de costes debe de tener las estimaciones mediante distintas técnicas, por ejemplo: Parkinson, Pricing to Win, juicio experto, puntos de función y puntos objeto, así como una comparativa de los resultados obtenidos mediante las distintas técnicas.

El presupuesto debe de tener:

- Los datos de proyecto, empresa y cliente (portada).
- Una descripción del producto a desarrollar (requerimientos).
- Los requisitos mínimos que necesita el programa (tanto SW como HW)
- Las disposiciones legales.
- Las retribuciones y plazos de pago.
- Podría contener un anexo donde se describan productos o servicios adicionales (HW, mantenimiento, migraciones, etc).

2.- ¿Los objetivos de la fase de construcción de un proyecto de que dependen? En un proyecto basado en UP y prototipado ¿Cuáles podrían ser los objetivos de la fase de construcción? Utilizando iteraciones de duración media ¿Cuántas iteraciones podríamos planificar detalladamente de la fase de construcción en el primer trimestre del proyecto? ¿Por qué? (2 p)

Dependen del problema, de modelo de proceso y del personal que tengamos. El problema (requerimientos funcionales y no funcionales) sugerirá una arquitectura que será uno de los condicionamientos más fuertes para determinar cómo vamos a construir el sistema.

Un ejemplo de objetivos de la fase de construcción podrían ser los que hemos tenido en prácticas: dotar de funcionalidad a las pantallas creadas en la fase de elaboración.

Suponiendo una duración media de 3 semanas por iteración, y suponiendo que tenemos sólo 1 iteración para elaboración, podríamos planificar 1 o 2. Lo normal sería no tener ninguna ya que la fase de elaboración suele tener más de 1 iteración.

3.- Indica las tareas que podemos realizar dentro de Microsoft Project (2 p). En el caso de un retraso en una actividad que pasos deberíamos de seguir para intentar minimizar el impacto en el proyecto.

En Microsoft Project podemos realizar las siguientes tareas.

- Crear el esquema de actividades WBS. Por ejemplo en prácticas hemos creado un WBS del modelo UP en tres niveles (fase, iteración, actividad).
- Asignar duraciones, precedencias y recursos al proyecto, es decir crear el plan temporal del proyecto.
- Crear la agenda, concretando las fechas de comienzo y fin de las tareas.
- Realizar el seguimiento del proyecto. Marcando hitos (objetivos) y haciendo su seguimiento comparando el plan real y el previsto (línea base) o con mediante datos cuantitativos (modelo EVA))

Los pasos que deberíamos e seguir para minimizar el impacto son:

- Consumir la holgura libre.
- Consumir la holgura total.
- Reasignar recursos.
- Quitar funcionalidades o retrasar el proyecto.

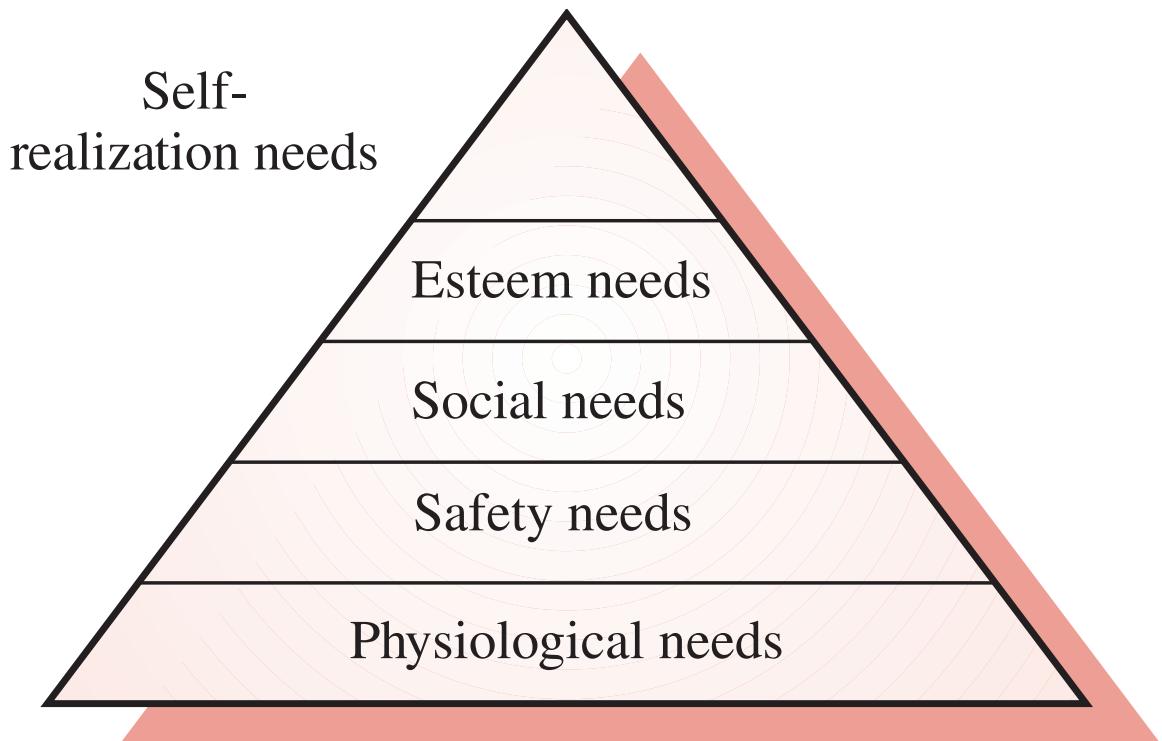
4.- Explica qué es la pirámide de necesidades. ¿Por qué es importante conocerla como gestores de proyectos? Pon un par de ejemplos de su uso. (2 p)

La pirámide de necesidades es una simplificación de la realidad, donde nos indica distintas necesidades que tenemos las personas. En la base de la pirámide están las necesidades principales que debemos cubrir: necesidades fisiológicas (comer, dormir, etc.), de seguridad, las necesidades ser aceptado como parte de un grupo, del reconocimiento de nuestro trabajo y las de autorealización.

Es importante como gestores de proyectos porque dentro de nuestras tareas están las relacionadas con el personal, en concreto las tareas de motivación y retención del personal serían las más vinculadas con la pirámide.

Como ejemplo de uso podríamos poner seleccionar a personal cuyas necesidades podamos cubrir a corto plazo y formar equipos donde sus necesidades sean satisfechas (si todos quieren lo mismo podría ser imposible satisfacer a todos).

Otro ejemplo sería el de retención, haciendo que el personal vaya cubriendo sus expectativas (necesidades) a lo largo de su vida laboral.



5.- Explica la diferencia entre mantenimiento y gestión de configuraciones. Explica la diferencia entre un gestor de versiones (como CVS, subversión, etc) y una base de datos de configuraciones. (2 p)

La gestión de configuraciones es una actividad de autoprotección para:

- ✓ Identificar el cambio
- ✓ Controlar el cambio
- ✓ Garantizar que el cambio se implementa adecuadamente
- ✓ Informar del cambio a todos aquellos que les interese

La gestión de configuraciones es una actividad que se lleva a cabo desde el principio del proyecto. El mantenimiento se lleva a cabo desde la puesta en producción del software. El mantenimiento hará uso de la gestión de configuraciones, podríamos decir que el mantenimiento es en cierta forma un subconjunto de la gestión de configuraciones. El mantenimiento incorpora unos procedimientos específicos teniendo en cuenta la puesta en marcha del software.

Un gestor de versiones como su nombre indica almacena las diferentes versiones de los elementos de configuración junto a cierta información muy rudimentaria (como la fecha/hora y un pequeño comentario del commit). Las herramientas como CVS y subversión están orientadas a gestionar versiones de los EC de código fuente. La base de datos de configuraciones es un término más ambicioso, en ella vamos a guardar todo tipo de información sobre el proceso de gestión de configuraciones de todo tipo de EC. La base de datos nos permitirá generar informes sobre todo el proceso de gestión de cambios.



GESTIÓN DE PROYECTOS INFORMATICOS

Convocatoria de JUNIO

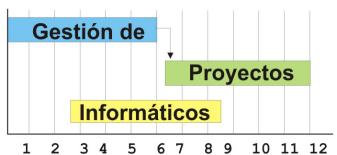
29 de MAYO de 2015

Apellidos, Nombre: _____

Grupo de prácticas: _____

1. Indica 5 factores que afecten a la estimación de costes de un proyecto. Propón una medida para ayudar a minimizar el impacto de cada uno de estos factores. (2,5p)
2. ¿Cuáles son las tareas que debemos de realizar como gestores de proyectos vinculadas con el personal? ¿Cómo puede afectar el personal del equipo al cono de la incertidumbre? Justifica las respuestas. (2,5p)
3. Crea un plan de riesgos para un proyecto ficticio (identifica sólo 3 riesgos). Indica los pasos seguidos y crea los documentos oportunos. Detalla como harías el seguimiento y control de dicho proyecto atendiendo a los riesgos identificados. (2,5p).
4. Dados los siguientes datos de un proyecto. Indica si son posibles los valores EVA de cada fila. Explica cuál es la situación de cada una de las tareas (si ha terminado, si va retrasada o no, y si se está ejecutando dentro del presupuesto). (2,5p)

Tarea	BCWS	BCWP	ACWP	BAC
Tarea 0	2800	1800	1800	2600
Tarea1	600	600	600	1200
Tarea2	1200	0	600	2600
Tarea3	1000	1200	800	2000
Tarea 4	0	0	0	600



GESTIÓN DE PROYECTOS INFORMATICOS

Call of June
May 29, 2015

Practices Group: _____

1. Write five factors which affect the estimated costs of a project. Propose a measure to minimize the impact of each of these factors. (2,5p)
2. What are the tasks we perform as project managers related to the staff? How can affect the members of the team to the uncertainty cone? Justify the answers (2,5p)
3. Create a risk plan for a fictitious project (identify just 3 risks). Indicate the steps followed and make the necessary documents. Detail how do you monitor and control the project in response to the risks identified. (2,5p).
4. Given the following project data. Indicate whether the EVA values in each row are possible. Explain the status of each task (if it has finished, if it is delayed or not, and if it is running within budget). (2,5p)

Task	BCWS	BCWP	ACWP	BAC
Task 0	2800	1800	1800	2600
Task 1	600	600	600	1200
Task 2	1200	0	600	2600
Task 3	1000	1200	800	2000
Task 4	0	0	0	600



GESTIÓN DE PROYECTOS INFORMATICOS

Convocatoria de JUNIO

29 de MAYO de 2015

Apellidos, Nombre: _____

Grupo de prácticas: _____

- Indica 5 factores que afecten a la estimación de costes de un proyecto. Propón una medida para ayudar a minimizar el impacto de cada uno de estos factores. (2,5p)

Factor	Description
Market opportunity	A development organisation may quote a low price because it wishes to move into a new segment of the software market. Accepting a low profit on one project may give the opportunity of more profit later. The experience gained may allow new products to be developed.
Cost estimate uncertainty	If an organisation is unsure of its cost estimate, it may increase its price by some contingency over and above its normal profit.
Contractual terms	A customer may be willing to allow the developer to retain ownership of the source code and reuse it in other projects. The price charged may then be less than if the software source code is handed over to the customer.
Requirements volatility	If the requirements are likely to change, an organisation may lower its price to win a contract. After the contract is awarded, high prices may be charged for changes to the requirements.
Financial health	Developers in financial difficulty may lower their price to gain a contract. It is better to make a small profit or break even than to go out of business

- Oportunidades de mercado: Realizar un proyecto empresarial, con un estudio claro de los costes y de la inversión, así como de la recuperación de los mismos. Crear proyectos diferenciados y mantener los secretos de nuestro software con el fin de evitar que la competencia entre en nuestro mercado.
- Incertidumbre en las estimaciones: Guardar datos de proyectos y trabajar para que en el futuro nuestras desviaciones del presupuesto sean menores. Estimaríamos con varias técnicas y compararíamos los resultados.
- Términos contractuales: Fijar los términos de forma clara mediante un contrato escrito.
- Volatibilidad de los requerimientos: Firmar un documento donde el cliente revise y acepte las especificaciones del sistema a construir.
- Salud financiera: Cobrar un anticipo en la firma del contrato. Guardar dinero de proyectos anteriores con el fin de tener autofinanciación total o parcial.

2. ¿Cuáles son las tareas que debemos de realizar como gestores de proyectos vinculadas con el personal? ¿Cómo puede afectar el personal del equipo al cono de la incertidumbre? Justifica las respuestas. (2,5p)

El gestor de proyecto tiene que realizar las siguientes actividades de gestión:

- Solución de problemas (usando la gente disponible).
- Motivación (de la gente que trabaja en el proyecto).
- Planificación (QUÉ tiene que hacer la gente).
- Estimación (CUÁN rápido tiene que trabajar la gente).
- Control (de las actividades de la gente).
- Organización (CÓMO tiene que trabajar la gente).

El personal puede afectar al cono de la incertidumbre haciendo que las estimaciones iniciales tengan mayor o menor desviación (incertidumbre).

- Si no tenemos al personal y hay que contratarlo, tendremos una mayor incertidumbre, ya que ni siquiera conocemos al equipo.
- Si es un equipo experimentado y si hemos trabajado con él en otras ocasiones tendremos una incertidumbre menor.

3. Crea un plan de riesgos para un proyecto ficticio (identifica sólo 3 riesgos). Indica los pasos seguidos y crea los documentos oportunos. Detalla como harías el seguimiento y control de dicho proyecto atendiendo a los riesgos identificados. (2,5p).

No encontramos/contamos con el personal necesario: Colaboramos con entidades académicas. Hacemos un proceso de selección con mayor alcance. Preveemos cursos de formación tanto para el personal nuevo como de reciclaje para los trabajadores actuales.

Personal se marche de nuestro proyecto: Incentivaríamos/motivaríamos al personal con el fin de retenerlo. Utilizaríamos modelos democráticos descentralizados donde todos hacemos de todo. Aplicaríamos alguna práctica como programación extrema (programación por parejas).

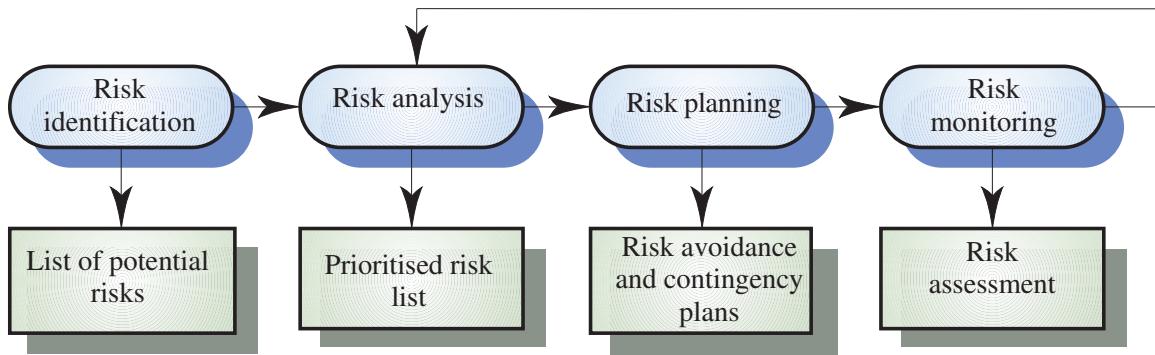
Infraestimación de costes: Guardar datos de proyectos y trabajar para que en el futuro nuestras desviaciones del presupuesto sean menores. Estimaríamos con varias técnicas y compararíamos los resultados. Aplicaríamos un margen de error a los presupuestos de coste/tiempo.

Habría que llenar el documento de identificación de riesgos, de análisis de riesgos, y planes de prevención y contingencia.....

El seguimiento lo haríamos para la falta de personal de la siguiente manera: Llevaremos un control de los conocimientos/cursos de nuestros empleados. Tendremos documentación de los planes de estudios actuales que utilizaremos para seleccionar las herramientas/personal.

El seguimiento ante el riesgo de que el personal se marche será viendo la evolución de trabajo, revisando las bajas/ausencias del personal y observando la actitud/estado del personal.

El seguimiento para la infraestimación de costes lo haremos haciendo un estudio EVA del coste real del proyecto. Este seguimiento debe de ser continuado a lo largo de todo el proyecto.



4. Dados los siguientes datos de un proyecto. Indica si son posibles los valores EVA de cada fila. Explica cuál es la situación de cada una de las tareas (si ha terminado, si va retrasada o no, y si se está ejecutando dentro del presupuesto). (2,5p)

Tarea	BCWS	BCWP	ACWP	BAC
Tarea 0	2800	1800	1800	2600
Tarea1	600	600	600	1200
Tarea2	1200	0	600	2600
Tarea3	1000	1200	800	2000
Tarea 4	0	0	0	600

Los datos de la actividad 0 son imposible ya que el BAC siempre tiene que ser mayor o igual que BCWS.

La tarea 1 ha comenzado y está realizada al 50%, va desarrollándose en el tiempo previsto y gastando también lo previsto.

La tarea 2 es imposible. No puede ser que no haya comenzado y haya consumido recursos.

La tarea 3 va adelantada (BCWP>BCWS) y consumiendo menos de lo previsto (ACWP>BCWP).

La tarea 4 aún no ha comenzado.



Ejercicios Repaso

Por si alguien abandona todos trabajamos en todo (utilizamos una estructura democrática).

Para detectar errores en las estimaciones iremos reestimando el proyecto en cada iteración. Preveremos unos márgenes de error para las dificultades encontradas. Debido al poco tiempo de desarrollo seleccionaremos las funcionalidades principales que necesita el cliente.

En el plan de contingencia se planifican las tareas para prevenir los riesgos y minimizar el impacto en el caso de que aparezcan los riesgos.

En el plan de seguimiento se van a establecer las actuaciones para detectar si un riesgo aparece o no.



Ejercicios Repaso

Cuestiones rápidas

1. Un plan de proyecto ...
 - a. Proporciona visibilidad al progreso del proyecto
 - b. Es una representación abstracta de un proceso que proporciona información parcial sobre el proceso software
 - c. Es un conjunto de actividades que tienen como objetivo el desarrollo de un producto software
 - d. Todas las afirmaciones anteriores son falsas

2. El modelo basado en prototipos ...
 - a. Reduce las horas de implementación.
 - b. Permite reducir riesgos y costes de desarrollo
 - c. El sistema resultante es un prototipo
 - d. Es posible utilizarlo sólo con lenguajes orientados a objetos

3. La gestión de proyectos ...
 - a. Tiene como objetivo determinar qué hay que hacer, cómo hay que hacerlo y quién tiene que hacerlo.
 - b. Está fuertemente condicionada por la implementación del sistema
 - c. Está fuertemente condicionada el diseño del sistema
 - d. Tiene como objetivo el confeccionar una agenda del proyecto

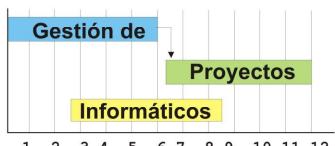
4. Tenemos tres actividades: A, B y C. A precede a B. Las holguras totales son: HT(A)= 4, HT(B)=2, HT(C)=3. Las holguras libres son: HL(A)= 3, HL(B)=2, HL(C)=3. Si A se retrasa 4 días, ¿qué les ocurre a B y C?
 - a. Nada, ya que no sobrepasamos la holgura total de A
 - b. La holgura total de B y C pasa a ser de 1 y 2 días respectivamente
 - c. B se convierte en crítica
 - d. Ninguna de las anteriores es cierta

5. Ms Project es ...
 - a. Una herramienta de seguimiento de proyectos
 - b. Una herramienta de modelado de proyectos
 - c. Una herramienta de estimación de costes de proyectos
 - d. Una herramienta de desarrollo de proyectos

6. El cono de la incertidumbre ...
 - a. Dice que conforme avanza el proyecto menos errores de estimación tenemos.
 - b. Dice que conforme avanza el proyecto más errores de estimación descubrimos.
 - c. No puedo aplicarlo sino hago el seguimiento del proyecto.
 - d. Todas las anteriores son falsas.

7. ¿Cuál de los siguientes elementos NO es un elemento de configuración (EC)?
 - a. Los ejecutables de la aplicación
 - b. La estructura de directorios.
 - c. La definición de las tablas de la BD de la aplicación
 - d. Todos los anteriores son elementos de configuración.

8. Make es una herramienta case que soporta ...
 - a. la planificación de una gestión de configuraciones
 - b. el control de cambios
 - c. la construcción del sistema
 - d. la gestión de versiones y releases



Ejercicios Repaso

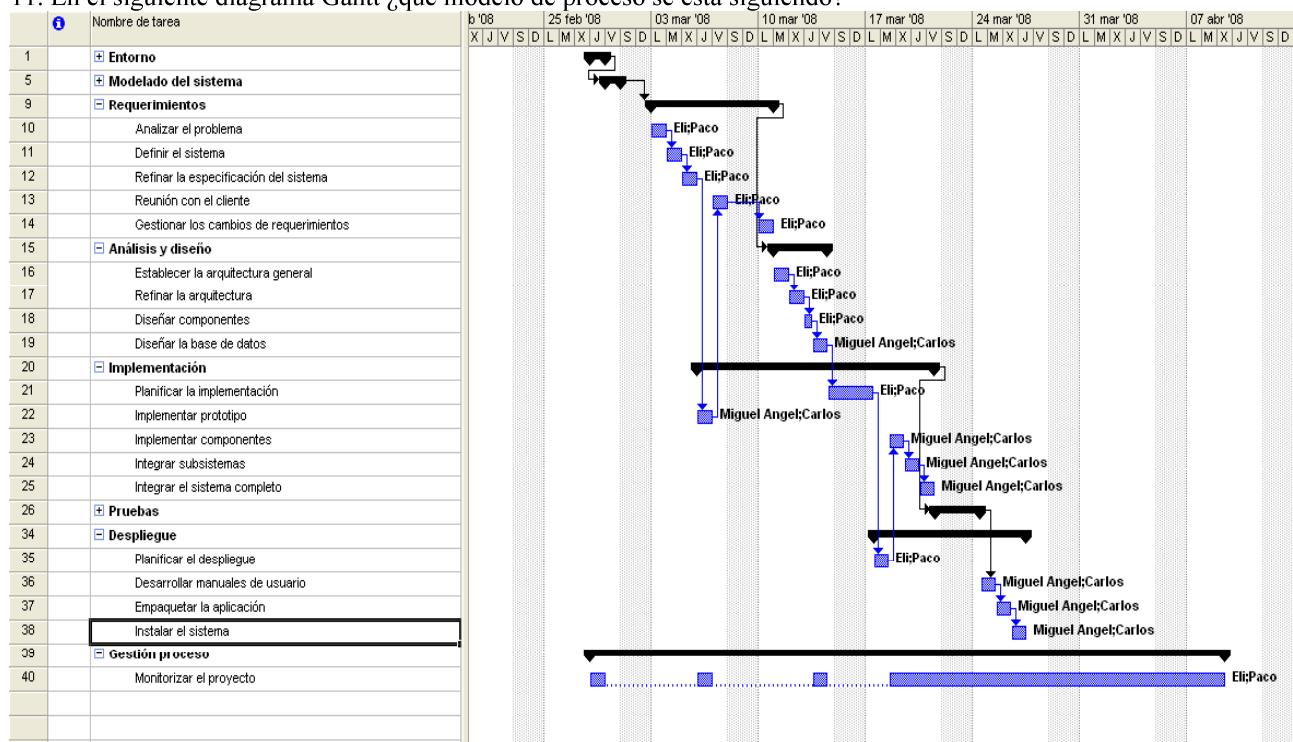
9. Las versiones de un sistema.

- a. Tienen un coste menor que las releases
- b. Suelen superar en número a las releases
- c. Incluyen documentación técnica
- d. Todas son ciertas.**

10. En el modelo UP.

- a. Hay una versión por iteración.
- b. Puede haber más de una versión por iteración.**
- c. Tenemos una release por fase.
- d. Todas las anteriores son ciertas.

11. En el siguiente diagrama Gantt ¿qué modelo de proceso se está siguiendo?



- a. Desarrollo Rápido de aplicaciones.
- b. Cascada
- c. Prototipado.
- d. Todos los anteriores.**

12. En la pirámide de jerarquía de necesidades donde se encuentran la mayoría de jefes de proyecto.

- a. No les afecta esta jerarquía, es sólo para empleados.
- b. Necesidades sociales.
- c. Necesidades seguridad.
- d. Necesidades autorrealización.**

13. ¿Cuál es la distribución de tiempo dentro de un equipo de desarrollo.

- a. 20% Trabajando sólo, 30% actividades no productivas, 50% Interactuando con otra gente.
- b. 30% Trabajando sólo, 20% actividades no productivas, 50% Interactuando con otra gente.**
- c. 30% Trabajando sólo, 20% actividades de gestión, 50% Interactuando con otra gente.
- d. 20% Implementando, 30% Interactuando con otra gente y 50% en actividades de gestión.



GESTIÓN DE PROYECTOS INFORMÁTICOS

Convocatoria de JUNIO. TEORIA

30 de mayo de 2017

Apellidos, Nombre: _____

Grupo de prácticas: _____

1. Explica que es un presupuesto y explica cuales son sus 3 objetivos. Propón una medida para cumplir con cada uno de estos objetivos. (2,5p)

Es un documento contractual entre el cliente y proveedor que sirve para establecer los productos y o servicios, la fecha de entrega del servicio así como el precio de los mismos. Los objetivos del presupuesto son: Vender, fortalecer las relaciones entre cliente y proveedor y crear oportunidad de negocio.

Para vender haríamos en presupuesto bien diseñado, con una presentación clara, información detallada y unos precios ajustados al mercado. Para fortalecer las relaciones haremos que los productos y o servicios estén bien definidos con el fin de evitar discrepancias futuras. Y para crear oportunidades de negocio ofreceremos servicios adicionales tales como hardware, migración datos, mantenimiento, formación.

2. Explica la diferencia entre WBS, plan general y plan detallado de un proyecto. ¿Qué diferencia existe entre el plan general y detallado en los modelos UP y cascada. (2,5p)

El WBS es un esquema de tareas estructuradas. El plan general fija las fases e iteraciones (siempre que el modelo tenga), así como sus objetivos. El plan detallado concreta las tareas, establece el orden y precedencias de las tareas y asigna duraciones y recursos a las tareas y establece las fechas.

La primera diferencia es que el modelo en cascada no tiene iteraciones. La segunda diferencia es que en el modelo en cascada el plan detallado se establece al principio del proyecto, mientras que en el modelo up se va detallando poco a poco. Otra diferencia son las fases, en cascada son análisis, diseño, implementación, integración y prueba, mientras que en el modelo UP son inicio, elaboración, construcción y transición.

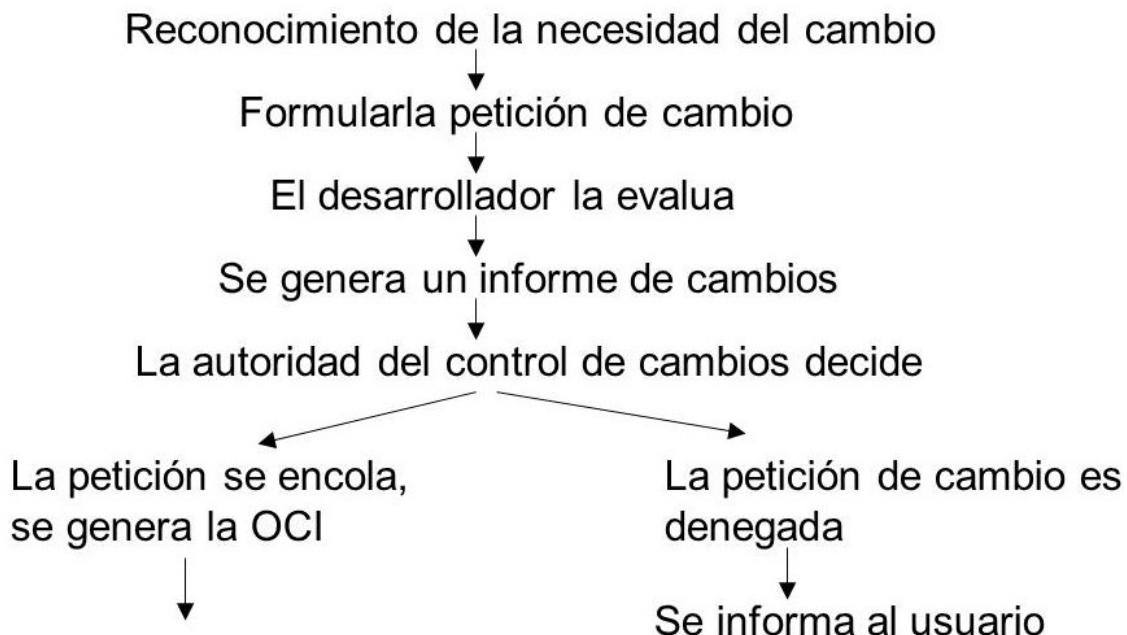
3. Describe 5 multiplicadores que nos afectan a las estimaciones de coste de un proyecto. (2,5p)

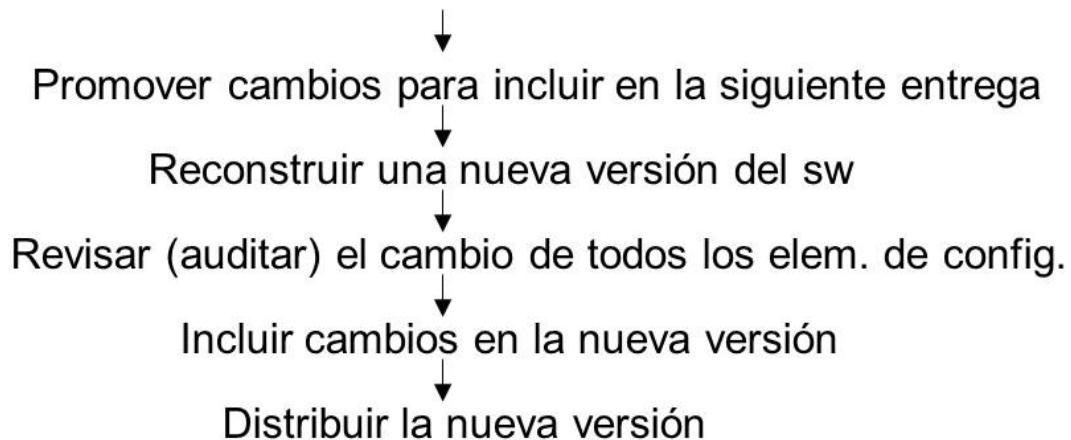
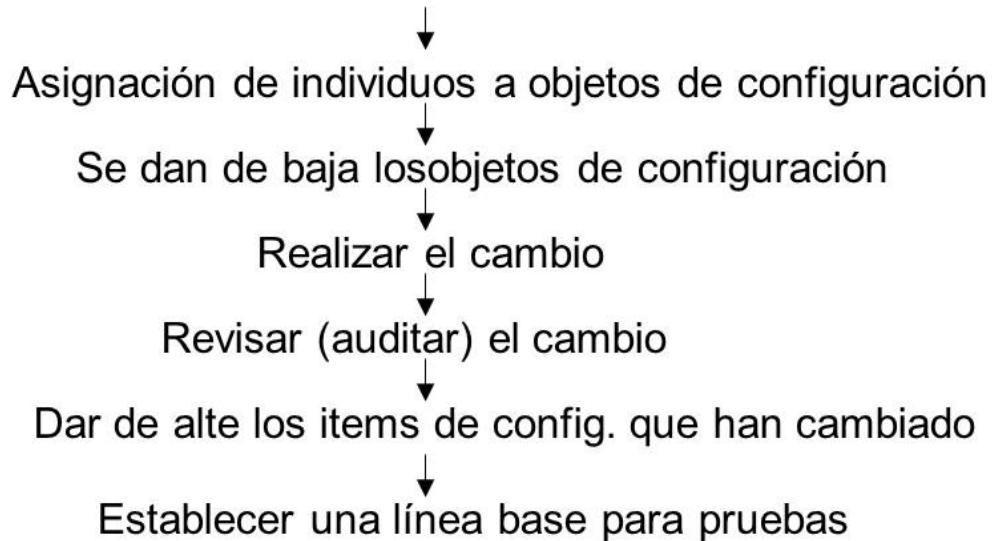
- RCPX - fiabilidad de producto y complejidad
- RUSE - reutilización requerida
- PDIF - dificultad de la plataforma
- PREX - experiencia del personal
- PERS - capacidad del personal
- SCED - agenda requerida
- FCIL - facilidades de soporte de grupo

Cuando un producto requiere **fiabilidad** su estimación de costes crecerá. Sin embargo si hablamos de **reutilización** tendremos un coste de integración de los componentes, pero siempre menor a la de su implementación. Si vemos que vamos a desarrollar con el fin de lo que desarrollemos lo aprovechemos (reutilicemos) en otros proyectos el coste sin embargo aumentará. Dependiendo de la **plataforma** los costes pueden variar, hay plataformas más sencillas, mejor preparadas para programar en ellas y con mayor tiempo en el mercado. Cuando hablamos del personal tenemos los términos de experiencia de capacidad. Lo interesante es que el personal tenga **experiencia**, disminuyendo así los costes del proyecto. El término **capacidad** va en la línea de no tener experiencia pero que los empleados sean capaces de asumir las nuevas tareas. Esto es muy típico en los ingenieros recién titulados, no tienen experiencia, pero si muchísima capacidad. Si hablamos de agenda poco flexible y apretada en el tiempo va provocar costes mayores.

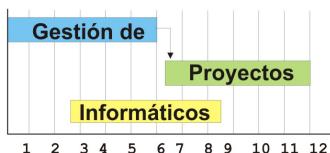
4. Explica la diferencia entre gestión de configuraciones y mantenimiento. Escribe el algoritmo de proceso de mantenimiento ¿En qué fase del proceso UP se realiza el mantenimiento?

La gestión de configuración es una actividad de autoprotección que se lleva a cabo a lo largo de todo el ciclo de vida del software. El mantenimiento es un subconjunto de la gestión de configuraciones que tiene lugar desde el momento que el software se pone en producción.





El modelo de proceso UP sólo abarca el desarrollo del software hasta su puesta en producción, no modela el mantenimiento.



GESTIÓN DE PROYECTOS INFORMÁTICOS

Convocatoria de JUNIO. PRÁCTICAS

30 de mayo de 2017

Apellidos, Nombre: _____

Grupo de prácticas: _____

- 1.- Elige un perfil profesional dentro de la informática y dibuja el diagrama de proceso de una competencia de dicho perfil. (0,75p)

Un perfil podría ser el técnico microinformático

La competencia reinstalar un ordenador de un cliente. En lugar del diagrama lo vamos a mostrar en forma de tabla, con entradas y salidas. En el diagrama de proceso representaríamos las acciones con círculos y los artefactos con rectángulos.

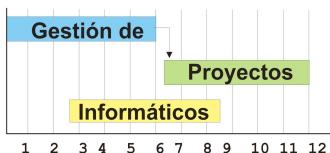
Entrada	Acción	Salida
	Recogida de datos del cliente	Listado de servicios, programas y configuraciones
Especificaciones fabricante	Comprobar requisitos mínimos	
Licencias aportadas por cliente	Comprobar licencias	
	Hacer imagen de la instalación actual	Imagen disco
	Verificar imagen	
	Exportar datos	Datos cliente
	Particionar disco	
	Instalar operativo	
Listado de servicios necesarios	Configurar servicios	
Listado de programas y configuraciones	Instalar y configurar programas	
	Verificar instalación	

2.- Explica la vinculación de la estimación realizada con Puntos Objeto con Pricing to Win.
(0,5p)

La técnica de Puntos Objeto estima el coste del proyecto, mientras que Pricing to Win estima el precio de venta. Al comparar ambos resultados nos dará el número de licencias de software que tenemos que vender para cubrir los costes. Esto implicará hacer un estudio de mercado para ver las ventas esperadas y el tiempo necesario para recuperar la inversión.

3.- ¿Qué dos formas hemos empleado en prácticas para realizar el seguimiento de un proyecto? ¿Qué ventajas presenta cada una de ellas? (0,75p)

Diagrama Gantt de Seguimiento y Modelo Eva. El Gantt de seguimiento permite realizar el seguimiento de una manera visual y rápida. El modelo Eva nos permite hacer un seguimiento objetivo y cuantitativo (con datos numéricos).



GESTIÓN DE PROYECTOS INFORMÁTICOS

Convocatoria de JUNIO. TEORIA

10 de julio de 2017

Apellidos, Nombre: _____

Grupo de prácticas: _____

1. Explica las diferencias entre un presupuesto y un documento de estimación de costes. ¿Qué información debe de contener cada uno de ellos? (2 p)

Las estimaciones de coste son un apartado del plan del proyecto, y es una información interna de la empresa. Indica el esfuerzo necesario para realizar un proyecto (personas/mes), el coste temporal y económico del proyecto.

El presupuesto es un documento que se va a entregar al cliente para informarle sobre los precios de venta, tiempos de entrega y funcionalidades de la aplicación.

La diferencia fundamental es que la estimación habla de costes, mientras que el presupuesto habla en términos de precios de venta (coste+margin error+beneficio) y fechas de entrega (entrega estimada+margin de error).

El documento de estimación de costes debe de tener las estimaciones mediante distintas técnicas, por ejemplo: Parkinson, Pricing to Win, juicio experto, puntos de función y puntos objeto, así como una comparativa de los resultados obtenidos mediante las distintas técnicas.

El presupuesto debe de tener:

Los datos de proyecto, empresa y cliente (portada).

Una descripción del producto a desarrollar (requerimientos).

Los requisitos mínimos que necesita el programa (tanto SW como HW)

Las disposiciones legales.

Las retribuciones y plazos de pago.

Podría contener un anexo donde se describan productos o servicios adicionales (HW, mantenimiento, migraciones, etc).

Dependen del problema, de modelo de proceso y del personal que tengamos. El problema (requerimiento funcionales y no funcionales) sugerirá una arquitectura que será uno de los condicionamientos más fuertes para determinar cómo vamos a construir el sistema.

Un ejemplo de objetivos de la fase de construcción podrían ser los que hemos tenido en prácticas: dotar de funcionalidad a las pantallas creadas en la fase de elaboración.

Suponiendo una duración media de 3 semanas por iteración, y suponiendo que tenemos sólo 1 iteración para elaboración, podríamos planificar 1 o 2. Lo normal sería no tener ninguna ya que la fase de elaboración suele tener más de 1 iteración.

2. ¿Qué unidades de medida se emplean para dar los datos de las estimaciones de costes? ¿Qué ventajas presentan estas unidades respecto a otras? ¿Qué unidades se emplean en los presupuestos? (2 p)

Todas las técnicas de estimación estiman el esfuerzo necesario para completar un proyecto. El esfuerzo se mide en **Personas/mes**.

Tiene la ventaja de permitirnos **comparaciones** entre proyectos de forma **atemporal** y entre **distintos lugares** (empresas/proyectos).

En los presupuestos utilizamos unidades **monetarias** (precio venta) y de **tiempo** (fecha entrega).

3.- Para un proyecto que se planifica con el modelo en cascada. ¿Qué estructura organizativa recomendarías?

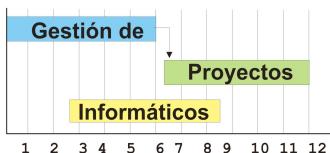
Dependiendo de las características del proyecto utilizare cualquiera de las 3 estructuras organizativas. Por ejemplo si es un proyecto con tamaño grande, dificultas baja y gran modularidad aplicaría el descentralizado controlado o el centralizado controlado.

		DD	DC	CC
DIFICULTAD	ALTA	X		
	PEQUEÑA		X	X
TAMAÑO	GRANDE		X	X
	PEQUEÑO	X		
DURACIÓN DEL EQUIPO	CORTO		X	X
	LARGO	X		
MODULARIDAD	ALTA		X	X
	BAJA	X		
FIABILIDAD	ALTA	X	X	
	BAJA			X
FECHA DE ENTREGA	EXTRICTA			X
	FLEXIBLE	X	X	
COMUNICACIÓN	ALTA	X		
	PEQUEÑA		X	X

4.- ¿Qué significa que tengamos un valor de BCWP menor que BCWS? ¿Qué pasos deberíamos de seguir en este caso, en qué orden y por qué? ¿Qué significa si tenemos un valor de ACWP mayor que BCWS? ¿Qué deberíamos de hacer en este caso? (2 p).

Si BCWP es menor que BCWS significa que el proyecto va retrasado. Lo primero que deberíamos de hacer es ver los indicadores de productividad. Si ACWP es menor que BCWP significa que no hemos utilizado los recursos previstos y habrá que aumentar la carga de trabajo a la tarea retrasada. Si ACWP es mayor o igual que BCWP en líneas generales podríamos restimar la duración de la tarea consumiendo su holgura libre, a continuación la holgura total, o si no es posible otra solución reasignar recursos o recortar incluso funcionalidades.

Comparar ACWP con BCWS no sirve para nada. Tendríamos que comparar con BCWP. En el caso de que ACWP sea menor que BCWP significa que nuestra productividad es buena (estamos gastando menos de lo presupuestado) y atendiendo al progreso, contrataremos a más gente si el este el lento, o no tendremos que aplicar acciones correctivas en caso contrario. En el caso de ser ACWP mayor que BCWP significa que estamos gastando más de lo presupuestado y tendremos que aplicar planes de contingencia relacionados con la mejora de la productividad.



GESTIÓN DE PROYECTOS INFORMÁTICOS

Convocatoria de JUNIO. PRÁCTICAS

10 de julio de 2017

Apellidos, Nombre: _____

Grupo de prácticas: _____

1.- ¿Qué son los formularios tipo? ¿Qué dos formas hay para registrar programas informáticos? (0,5p)

Los formularios tipos son formularios simplificados (que llevan los datos básicos rellenados) para dar de alta ficheros en la agencia de protección de datos.

Las dos formas son a través de patente y de derechos de autor. La primera de ellas se realizaría a través del ministerio de industria y es más complejo. La segunda es mucho más sencilla y se hace a través del ministerio de cultura y deporte.

2.- ¿Qué tipos de planes hemos hecho en prácticas? Justifica la respuesta. (0,75)

Hemos hecho un plan general al principio del proyecto y un plan detallado de las 2 primeras iteraciones. En el modelo UP al principio del proyecto no tenemos toda la información (es un modelo iterativo), por lo tanto sólo desarrollamos un plan general. Conforme avanza el proyecto vamos concretando el plan, detallando una serie más de iteraciones.

3.- ¿Qué diferencias hay entre un plan de 3 y 4 niveles? ¿Qué estructura de niveles hemos utilizado en prácticas? ¿Cómo hemos solventado los problemas que presenta esa estructura? (0,75)

En un plan en 3 niveles tenemos los niveles de fase, iteración y tarea. En un plan de 4 tenemos fase, iteración, disciplina y actividad. En prácticas hemos utilizado el de 3 ya que presenta una mejor visibilidad. Para evitar sus desventajas hemos utilizado un esquema de nombrado (verbo+sustantivo) y una tormenta de ideas. La primera medida iba orientada a saber exactamente qué íbamos a hacer y de qué disciplina se trataba. La segunda era para no olvidar tareas, o al menos no dejar una disciplina sin actividades.