

## Administración de Proyectos de Software

### Planificación de Proyectos

E. Estévez - P. Fillottrani

Depto. Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur

segundo semestre 2019



## Planificación de Proyectos

Sesión de Planificación

WBS

Estimación de Esfuerzos

Diagramas de Red del Proyecto

COCOMO

Propuesta de Proyecto



## Importancia de la Planificación

- ▶ para el LP, no es sólo una ruta para el trabajo a realizar, sino también es un ayuda en la toma de decisiones
- ▶ es esperable el cambio, no hay que tomar la planificación como algo fijo
- ▶ hay que estar preparado para el cambio
- ▶ beneficios:
  - ▶ **reducción de incertidumbre**
  - ▶ **incremento del conocimiento**
  - ▶ **mejora en la eficiencia**



## Uso de herramientas de Software

- ▶ el tamaño del proyecto influye en el tipo de herramientas a usar
- ▶ **ventaja:** permiten investigar distintas alternativas ajustando parámetros
- ▶ **desventaja:** no hay sido muy efectivos en el manejo de recursos y a veces producen resultados desconcertantes
- ▶ **desventaja:** costos (dinero, tiempo de aprendizaje y de mantenimiento)
- ▶ es muy importante incorporarles **estimaciones confiables**



## Tiempo para la Planificación

- ▶ proyectos tipo D menos de un día
- ▶ proyectos tipo C hasta dos días
- ▶ proyectos tipo B tres o cuatro días
- ▶ proyectos tipo A cinco o más días



## Sesión Conjunta de Planificación de Proyecto (JPPS)

- ▶ una sesión grupal para generar el plan detallado
- ▶ **objetivo** desarrollar un plan que satisfaga el COS y describa el POS
- ▶ **obstáculo** poco tiempo de los miembros para planificar, reuniones virtuales



## Herramientas y Procesos

- ▶ JPPS (Joint Project Planning Session)
- ▶ WBS (Work Breakdown Structure)
- ▶ técnicas de estimación de duración, recursos y costos
- ▶ técnicas de construcción y compresión de cronogramas
- ▶ planificación del manejo de riesgos
- ▶ Propuesta de Proyecto
- ▶ obtención de la aprobación de la propuesta



## Agenda de la JPPS

- ▶ revisión del POS
- ▶ creación del WBS
- ▶ estimación de duración y recursos para tareas
- ▶ construcción del diagrama de red del proyecto
- ▶ determinación del camino crítico
- ▶ revisión y aprobación de la fecha de terminación
- ▶ completar el cronograma
- ▶ obtener consenso sobre el plan



## Roles

- ▶ **facilitador** conduce la sesión, de mente abierta (no el LP) y con experiencia
- ▶ **LP** se concentra en el plan y no en la reunión, debe sentirse confiado en el plan resultante
- ▶ **consultor JPPS** sirven como expertos externos para opinar sobre aspectos específicos, especialmente si no hay personal con experiencia
- ▶ **tecnógrafo** ayuda al facilitador con el uso de las herramientas de software para documentar el proyecto
- ▶ **equipo central del proyecto** participantes senior (que ya hayan sido involucrados), desde el primer día hasta el último



## Roles

- ▶ **representante del cliente** puede ser problemático
- ▶ **administradores de recursos** expertos en estimación
- ▶ **campeón del proyecto** es el encargado de “vender” el proyecto al/los gerentes. Puede ser el cliente
- ▶ **administradores funcionales** son los que proveen inputs o reciben outputs del proyecto. Aseguran la integración de los entregables



## Entregables de la JPPS

- ▶ WBS, incluyendo estimación de duración y recursos
- ▶ Diagrama de Red del proyecto
- ▶ Cronograma de Actividades
- ▶ Asignación de Recursos
- ▶ Documento del Proyecto y Propuesta del Proyecto

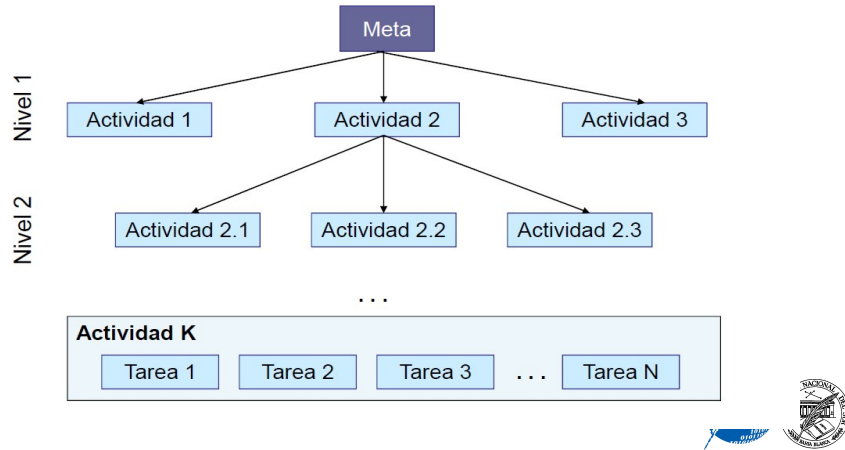


## Work Breakdown Structure

- ▶ el **WBS** es una descripción jerárquica del trabajo que se debe realizar para completar el proyecto
- ▶ es similar a una descomposición funcional
- ▶ el trabajo se divide en actividades
- ▶ las actividades se dividen en tareas
- ▶ es la convención que utilizaremos, otros autores la intercambian



## Gráfico WBS

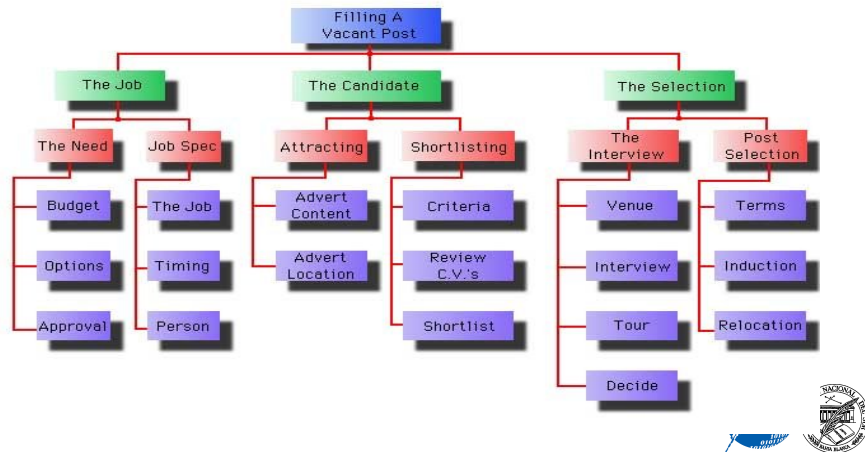


## Usos del WBS

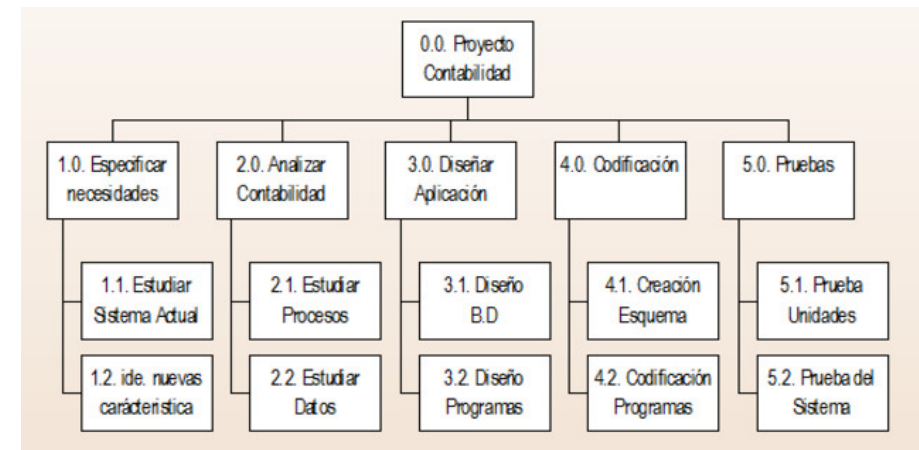
- ▶ **diseñar y planificar el trabajo:** permite a los integrantes del equipo visualizar cómo puede definirse y administrarse el trabajo del proyecto
- ▶ **diseñar la arquitectura:** es un gráfico del trabajo del proyecto, muestra cómo se relacionan los distintos ítems de trabajo a realizar
- ▶ **planificar:** se debe estimar esfuerzo, tiempos, y recursos para el último nivel
- ▶ **informar el estado del proyecto:** es usada como una estructura para mostrar el grado de avance



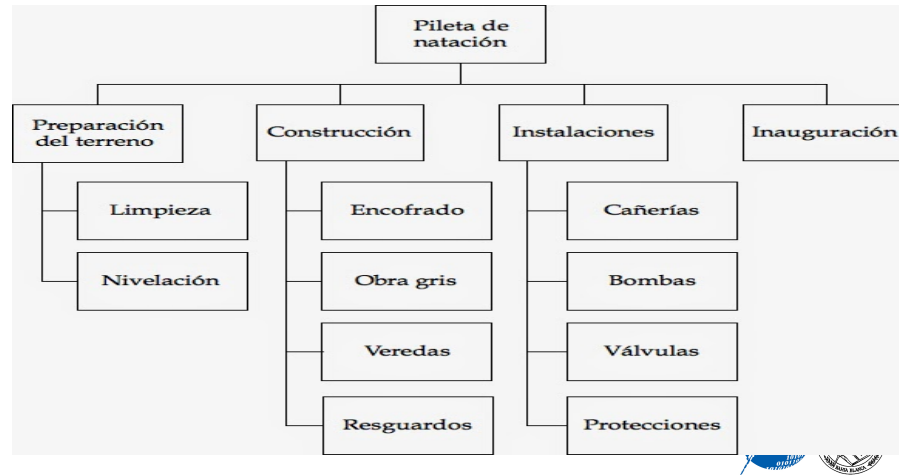
## Ejemplo de WBS



## Ejemplo de WBS



## Ejemplo de WBS

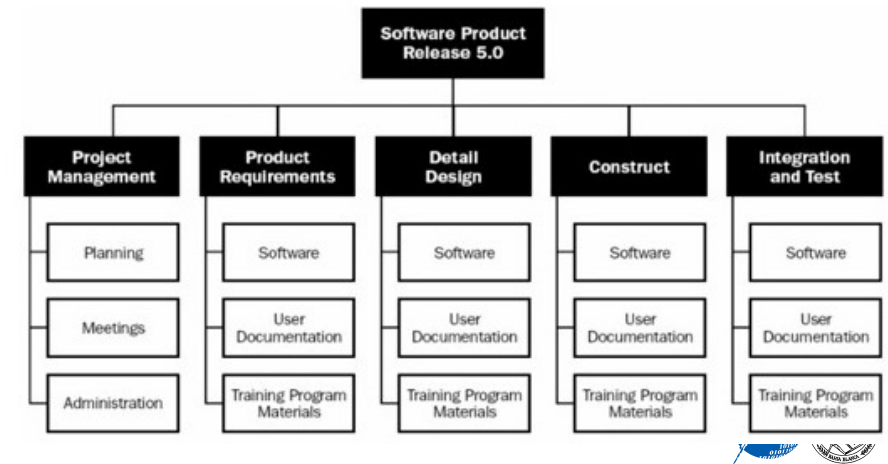


## Construcción del WBS

- ▶ su confección es responsabilidad del LP
- ▶ debe definirse de tal manera que el LP pueda administrar el proyecto
- ▶ manera de que funcione: stickers, resaltadores, pizarrón, paredes.
- ▶ formas de construirlo:
  1. top-down
    - 1.1 equipo completo
    - 1.2 sub-equipos
  2. bottom-up



## Ejemplo de WBS



## Top-down – equipo completo

- ▶ todos los miembros del equipo participan de la descomposición
- ▶ se comienza con el nivel 0 (el de la meta) y se particiona sucesivamente hasta que los participantes estén satisfechos de que el trabajo ha sido suficientemente definido
- ▶ debido a que las tareas se definen con el suficiente nivel de detalle, las estimaciones de costo, tiempo y recursos son más exactas
- ▶ una vez que las tareas se han definido, se deben secuenciar. Se debe analizar qué actividades o tareas se pueden hacer concurrentemente
- ▶ **ventaja:** brinda la oportunidad de que todos presten atención al WBS, y se discuta en el momento



## Top-down – subequipos

- ▶ el equipo completo acuerda la partición del primer nivel
- ▶ se crean tantos sub-equipos como actividades haya en el nivel uno
- ▶ cada sub-equipo particiona una actividad (se le asigna la actividad para la cual tenga más experiencia)
- ▶ un sub-equipo puede solicitar ayuda externa
- ▶ demanda menos tiempo que el enfoque anterior



## Bottom-up

- ▶ se asemeja a una lluvia de ideas (brainstorming)
- ▶ el grupo clasifica las tareas que parecieran relacionarse
- ▶ se reúnen todos los grupos y cada grupo presenta sus resultados. Se discute en conjunto
- ▶ la desventaja de este enfoque es no definir las tareas con el suficiente grado de granularidad
- ▶ existen metodologías que ayudan a la descomposición de actividades: proveen listados de las tareas



## Descripción completa de una tarea

1. estado medible
2. acotada
3. producir un entregable
4. tiempo y costo estimable
5. duración aceptable
6. independiente



## Completitud - Estado medible

- ▶ **el estado de una tarea debe ser medible**
- ▶ en cualquier momento se debería poder determinar el estado en que se encuentra
- ▶ ejemplo: codificar 10 componentes. Tiempo asignado: 10 semanas.
- ▶ luego de 4 semanas el LP pregunta cuál es el estado de la tarea. ¿Cuáles serían las respuestas válidas?



## Completitud - Acotada

- ▶ una tarea debe ser acotada
- ▶ debe poseer:
  - ▶ evento de comienzo – fecha de comienzo
  - ▶ evento de fin – fecha de fin



## Completitud - Tiempo y Costo Medible

- ▶ una tarea debe tener un tiempo y un costo medibles
- ▶ el tiempo y costo deben ser fácilmente estimables
- ▶ realizar la estimación de tiempo y costo para las tareas de menor nivel, permite luego agregar y calcular el costo y tiempo total del proyecto



## Completitud - Producir un Entregable

- ▶ una tarea debe producir un entregable
- ▶ el entregable es un signo visible de que la tarea se completó. Puede ser:
  - ▶ un producto
  - ▶ un documento
  - ▶ una autorización para continuar con la próxima tarea
  - ▶ etc.



## Completitud - Duración Aceptable

- ▶ la duración de una tarea debe ser aceptable
- ▶ en lo posible no trabajar con tareas de más de 10 días - 2 semanas laborales
- ▶ cuidado: hay excepciones



## Completitud - Independiente

- ▶ una tarea debe ser independiente
- ▶ es importante la independencia de tareas
- ▶ una vez que se comenzó una tarea se debe poder continuar razonablemente sin interrupciones y sin la necesidad de un input adicional
- ▶ el esfuerzo dedicado debe ser continuo

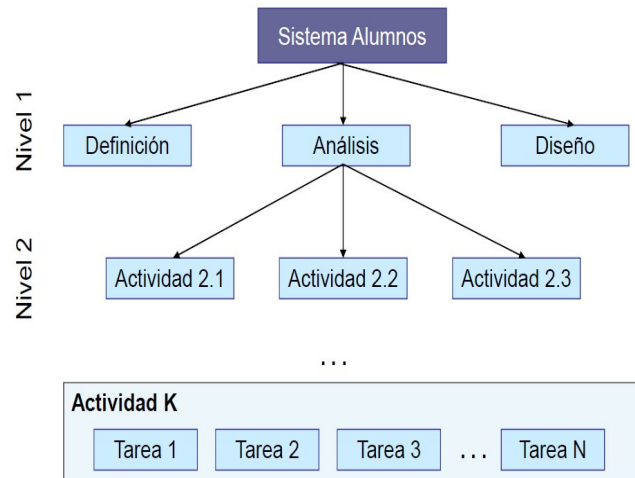


## Enfoques para la construcción

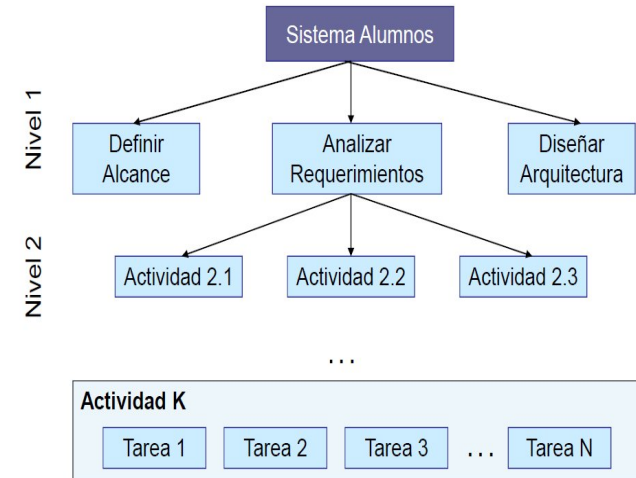
- ▶ no hay reglas fijas
- ▶ se pueden estipular criterios para nombrar las tareas:
  - ▶ enfoque por sustantivos: en función de los entregables
  - ▶ enfoque por verbos: en función de las acciones requeridas para producir el entregable
  - ▶ enfoque organizacional: en función de las unidades organizativas que trabajarán en el proyecto



## Construcción - por sustantivos

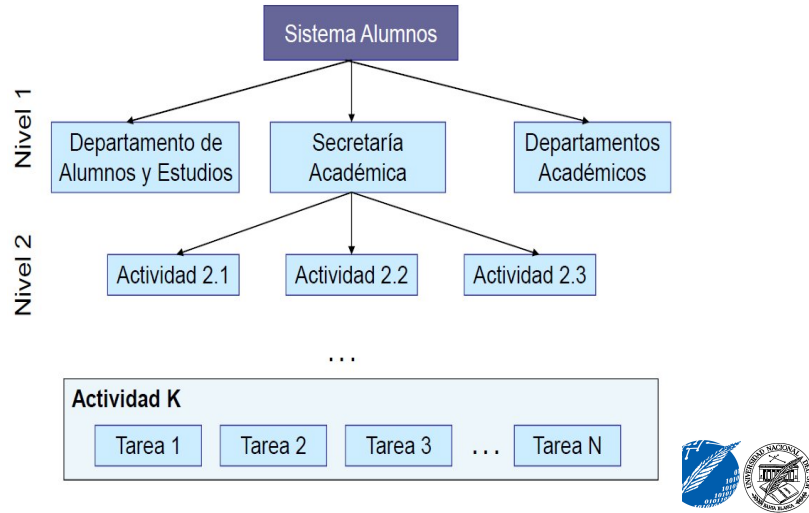


## Construcción - por verbos





## Construcción - organización



## Estimación de la Duración - Técnicas

- ▶ similitud con otras tareas
- ▶ datos históricos
- ▶ juicio experto
- ▶ técnica Delphi
- ▶ técnica de 3 puntos
- ▶ técnica Delphi de banda ancha



## Similitud con Otras Tareas

- ▶ estimar en base a las estimaciones de tareas similares de otros proyectos
- ▶ los datos están en la memoria de las personas



## Datos Históricos

- ▶ estimar en base a las estimaciones de tareas similares de otros proyectos
- ▶ los datos están en un registro – base de datos, no sólo en la memoria de las personas
- ▶ la base de datos histórica puede ser tan sofisticada como se desee



## Juicio Experto

- ▶ las estimaciones las realizan consultores externos o vendedores con experiencia en la metodología o en la tecnología
- ▶ si el juicio experto se basa en la estimación de vendedores, las estimaciones pueden no ser objetivas



## Técnica Delphi

- ▶ es una técnica de grupo que extrae y resume el conocimiento del grupo para arribar a una estimación
- ▶ se le pide a cada miembro del grupo a que realice su estimación
- ▶ se tabulan los resultados **Primera Pasada**



## Técnica Delphi



## Técnica Delphi

- ▶ aquellos participantes cuyas estimaciones cayeron en los cuartiles exteriores, se les pide que justifiquen su estimación
- ▶ luego de escuchar los argumentos, se les pide a los miembros que vuelvan a estimar
- ▶ los resultados se presentan en un histograma rotulado **Segunda Pasada**
- ▶ las posiciones extremas se defienden



## Técnica Delphi

- ▶ nuevamente se hace otra estimación
- ▶ los resultados se publican como la **tercera pasada**
- ▶ se permiten ajustes finales
- ▶ el promedio de la tercera pasada se usa como estimación del grupo



## Técnica de 3 Puntos

- ▶ se necesitan 3 estimaciones de la duración de la tarea: **optimista**, **pesimista**, **media**
- ▶ **estimación optimista**: es la duración mas corta suponiendo que todo suceda de acuerdo a la planificado
- ▶ **estimación pesimista**: la duración de la tarea suponiendo que falle todo lo que se prevee que puede fallar
- ▶ **estimación media**: la duración normal (usual) de la tarea
- ▶ se obtiene

$$estimacion = (optimista + 4 * media + pesimista) / 6$$



## Técnica Delphi de Banda Ancha

- ▶ es una combinación de la técnica Delphi y la de 3 puntos
- ▶ se basa en la técnica Delphi pero a cada integrante se le pide que haga las 3 estimaciones: la optimista, la pesimista y la media
- ▶ se recopilan los resultados y se eliminan los extremos
- ▶ se calculan los promedios de optimistas, pesimistas y medias
- ▶ se calcula con la fórmula de 3 Puntos utilizando los promedios



## Otros Métodos de Estimación

- ▶ otros autores proveen la siguiente clasificación de los métodos de estimación:
  - ▶ analogía con proyectos previos
  - ▶ estimar tamaño
    - ▶ longitud (LOC)
    - ▶ funcionalidad (Puntos de Función de Albrecht)
    - ▶ modelos algorítmicos:  $a * (KLOC)^b * FactorAjuste[CoCoMo]$
  - ▶ estimación top-down o bottom-up



## Duración

- ▶ **duración** es el tiempo transcurrido en días laborables para finalizar el proyecto - sin considerar feriados, fines de semana, días no laborables
- ▶ **esfuerzo de trabajo**: es la labor requerida para completar una tarea. La labor se puede realizar en horas consecutivas o no
- ▶ la duración es diferente al esfuerzo de trabajo. Duración: 10 días - Esfuerzo de trabajo: 20 horas
- ▶ el **tiempo transcurrido** es diferente al **tiempo de trabajo** en una tarea. Existen imprevistos, interrupciones, actividades sociales.



## Carga de Recursos

- ▶ la duración de una tarea es **influenciada** por la cantidad de recursos planificados para trabajar en ella
- ▶ se dice influenciada, ya que no es una relación lineal directa entre la cantidad de recursos asignados a la tarea y la duración de la misma
- ▶ **crash de la tarea**: agregar más recursos para mantener la duración de una tarea dentro de los límites planificados
- ▶ ejemplo: traslado de la silla con una persona y con dos personas



## Carga de Recursos

- ▶ **crashpoint de la tarea**: es el punto en el cual agregar más recursos aumenta la duración de la tarea
- ▶ ejemplo: traslado de la silla con cuatro personas
- ▶ el agregar n personas a una tarea, hace que se agreguen:
  - ▶ como mínimo n canales de comunicación más
  - ▶ trabajo de coordinar a estas personas
  - ▶ nuevas tareas (capacitación, supervisión, coordinación)
- ▶ otra consideración para el LP al agregar recursos a una tarea es considerar el impacto del riesgo de esta decisión. Ej: distintos enfoques de trabajo, mayor probabilidad que alguien tenga problemas, ...



## Variantes en la Duración

- ▶ existen distintas causas por las variantes en la duración de una tarea:
  - ▶ **diferencias en los perfiles**
  - ▶ **eventos inesperados**
  - ▶ **eficiencia del tiempo de trabajo**
  - ▶ **errores e interpretaciones erróneas**



## Variantes - Diferencias en los perfiles

- ▶ la estrategia es estimar la duración de la tarea basados en personas con un determinado perfil
- ▶ las personas asignadas pueden tener distintos perfiles y esto implica cambios en la duración



## Variantes - Eventos Inesperados

- ▶ demoras de proveedores
- ▶ fallas de energía
- ▶ incorrecto envío de materiales
- ▶ enfermedades
- ▶ problemas técnicos



## Variantes - Eficiencia del trabajo

- ▶ cada vez que un trabajador es interrumpido, le demanda más tiempo volver al nivel de productividad previo al momento de la interrupción
- ▶ se logra mayor eficiencia al realizar trabajos de manera focalizada
- ▶ algunas personas se ven mas afectadas que otras



## Variantes - Errores y malentendidos

- ▶ existen errores e interpretaciones erróneas sobre los trabajos a realizar
- ▶ esto puede implicar rehacer trabajo ya hecho



## Clasificación de Recursos

- ▶ **humanos** son los más difíciles de estimar en un proyecto de software
- ▶ **facilidades** son lugares físicos (salas de reunión, salas de conferencias). Se planifican las fechas y el tiempo que se utilizarán
- ▶ **equipamiento** es tratado como otras facilidades. Ej: proyector, notebook
- ▶ **viáticos** gastos por representación, refrigerios
- ▶ **materiales**: papel, tinta, CDs, DVDs, etc.



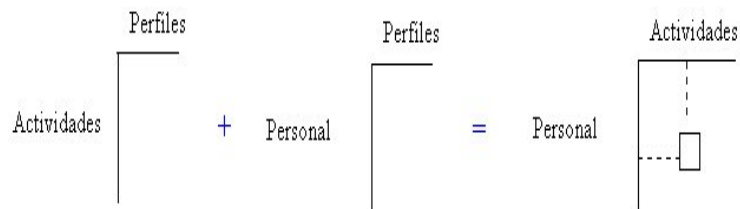
## Personas como Recursos

- ▶ los recursos personas se asignan por perfil y no con nombre y apellido
- ▶ una vez definido el rol, se personaliza y ahí es donde se presentan los problemas
- ▶ hay pocas herramientas para ayudar a planificar a las personas
- ▶ una herramienta es la **Matriz por Perfil**



## Matriz por Perfil

- ▶ se confecciona una matriz por **Actividad-Perfil** y otra matriz por **Persona-Perfil**
- ▶ se componen las matrices



## Duración y Disponibilidad de Recursos

- ▶ existen tres variables que influyen en estimar la duración de una tarea:
  - ▶ la duración en sí misma
  - ▶ el esfuerzo de trabajo que se le dedicará a la tarea. Ej: horas, o días de una persona
  - ▶ el porcentaje de tiempo que la persona le dedica a la tarea diariamente
- ▶ todo software de administración de proyectos permite cargar 2 de estas variables y calcula la tercera automáticamente.



## Cálculo

► Formas de Cálculo:

$$\text{esfuerzo} / \text{porcentaje} = \text{duración}$$

- se puede asignar *esfuerzo* y *porcentaje*
- o se puede asignar *duración* y *esfuerzo*
- o se puede asignar *duración* y *porcentaje*
- o puede haber una asignación variable



## Cálculo

► Formas de Cálculo:

$$\text{esfuerzo} / \text{porcentaje} = \text{duración}$$

- asignar el esfuerzo de trabajo requerido y un porcentaje de trabajo asignado a la actividad por día:
  - esfuerzo requerido: 40 horas
  - la persona dedica el 50% de su tiempo
- ¿cuál es la duración total?
- 

$$40 \text{ horas} / 0,50 = 80 \text{ horas}$$



## Cálculo

► Formas de Cálculo:

$$\text{esfuerzo} / \text{porcentaje} = \text{duración}$$

- asignar la duración total y el esfuerzo de trabajo requerido
  - esfuerzo requerido: 5 días de una persona
  - la duración total es de 10 días
- ¿cuál es el porcentaje que se debe dedicar a la tarea?
- 

$$5 \text{ dias} \times \text{pers} / 10 \text{ dias} = 0,5 \text{ pers}$$



## Cálculo

► Formas de Cálculo:

$$\text{esfuerzo} / \text{porcentaje} = \text{duración}$$

- asignar la duración total y un porcentaje de trabajo asignado a la actividad por día
  - la duración total es de 10 días
  - la persona dedica el 50% de su tiempo
- ¿cuál es el esfuerzo requerido?
- 

$$10 \text{ dias} * 0,5 \text{ pers} = 5 \text{ dias} \times \text{pers}$$



## Cálculo

- ▶ Formas de Cálculo:

$$\text{esfuerzo} / \text{porcentaje} = \text{duración}$$

- ▶ el porcentaje de trabajo asignado a la actividad por día es variable
  - ▶ el esfuerzo de trabajo es de 50 horas
  - ▶ la persona trabaja al 50% las primeras 10 horas y luego al 25%
- ▶ ¿cuál es la duración total de la actividad?

$$10 \text{ horas} \times \text{pers} / 0,50 \text{ pers} + 40 \text{ horas} \times \text{pers} / 0,25 \text{ pers} = 180 \text{ horas}$$

- ▶ se debe evitar, particionando las actividades



## Estimación de Costos

- ▶ predicciones de cuanto tiempo, esfuerzo y perfiles de RRHH son requeridos para construir un sistema de software
- ▶ muchas veces se intercambia estimación de esfuerzo con estimación de costos
- ▶ las estimaciones preliminares son las más difíciles y las menos exacta
- ▶ en Ingeniería de Software somos notoriamente inexactos para calcular tiempo y costo



## Estimación de Costos

- ▶ a diferencia de otras profesiones donde se puede tomar ventaja de las tareas repetitivas, esto no ocurre en IS
- ▶ diferencias:
  - ▶ dominio de aplicación
  - ▶ hardware
  - ▶ herramientas
  - ▶ técnicas
  - ▶ personal



## Estimación de Costos

- ▶ en IS somos mas **creadores** que constructores
- ▶ problemas de estimación:
  - ▶ **problemas políticos**: cuando las estimaciones se convierten en objetivos, cuando se ajusta el precio por conveniencia
  - ▶ **problemas técnicos**: No existen datos históricos para estimar





## Usos de la Estimación de Costos

- ▶ la **estimación de costos** es parte del planeamiento de cualquier actividad de ingeniería
- ▶ la diferencia en IS es que el costo principal son los recursos humanos
- ▶ la estimación de costos tiene dos usos:
  - ▶ en planificación: se necesita saber cuantos recursos va a insumir
  - ▶ en control: se necesita saber cuanto se hizo y cuanto falta
- ▶ se necesitan **métodos predictivos** para estimar la complejidad del software antes de que sea desarrollado



## Estimación de Costos

- ▶ teniendo la estimación del tiempo y de los recursos requeridos, se puede comenzar la **estimación de costos**
- ▶ para evaluar el costo, se asigna el costo de una unidad del ítem por la duración
- ▶ la estimación de costos comienza a nivel de tarea
- ▶ una vez que se han determinado los costos de todas las tareas y actividades, se agregan a nivel de proyecto
- ▶ para simplificar, separamos los costos en: laborales y de materiales



## Estimación de Costos

- ▶ los **costos de materiales** están dados por los otros recursos que no son personas
- ▶ los costos de materiales se estiman por precio o por uso
- ▶ los **costos laborales** se calculan por perfil
- ▶ se necesita conocer el valor horario para cada perfil



## Estimación de Costos Laborales - ejemplos

<http://www.cpciba.org.ar>  
<http://www.cpcipc.org.ar>  
<http://www.encuestasit.com/Sueldo.aspx>



## Diagramas de Actividades del Proyecto

- ▶ una vez definidas las actividades (construido el WBS), se debe decidir en qué orden se realizarán
- ▶ con las actividades y la duración, se puede construir un gráfico que represente el modelo del proyecto
- ▶ de este gráfico se desprenderán datos importantes para la administración del mismo:
  - ▶ el tiempo más temprano de comienzo de una tarea
  - ▶ el tiempo más temprano que se podrá finalizar el proyecto
- ▶ **diagrama de red del proyecto**: es una representación de la secuencia en la que se realizarán las tareas del proyecto



## Diagramas de Actividades del Proyecto

- ▶ existen dos formas de representar esta misma información:
  1. **diagrama de Gantt**
  2. **diagrama de Red** que a su vez se pueden describir según
    - 2.1 **método AON** activity on the node PDM precedence diagramming method
    - 2.2 **método AOA** activity on the arrow PERT program evaluation and review technique



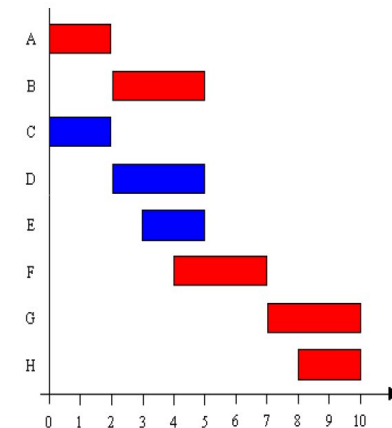
## Diagramas de Gantt

- ▶ desarrollado por Henry L. Gantt como técnica de control de proyectos que puede ser usada para varios propósitos:
  - ▶ planificación temporal
  - ▶ presupuestos
  - ▶ planificación de recursos
- ▶ características:
  - ▶ se representa mediante un cuadro de doble entrada. Es un diagrama de barras
  - ▶ cada barra representa una actividad, medios de producción, recursos
  - ▶ se dibujan sobre un eje de tiempo
  - ▶ la longitud de la barra indica el tiempo requerido



## Diagramas de Gantt - ejemplo

Tarea	Predec.	Duración
A	-	2
B	A	3
C	-	2
D	C	3
E	D <sub>II+1</sub>	2
F	B <sub>F1-1</sub>	3
G	D, E, F	3
H	G <sub>FF</sub>	2



## Diagramas de Gantt - evaluación

- ▶ **ventajas**
  - ▶ es muy sencillo de construir
  - ▶ no requiere ninguna herramienta automatizada específica
- ▶ **desventajas**
  - ▶ no contiene información detallada. Sólo indica las tareas, fechas de comienzo y fin
  - ▶ no provee información sobre la manera en que se puede terminar anticipadamente el proyecto o si la asignación de recursos es la más efectiva

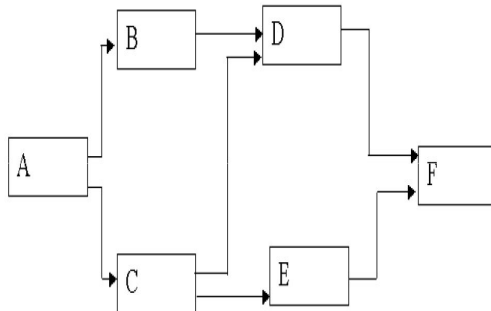


## Diagramas de Gantt - construcción

1. dibujar los ejes horizontal y vertical. Escribir los nombres de las tareas sobre el eje vertical, y el tiempo sobre el horizontal
2. dibujar los bloques correspondientes a las tareas que no tienen predecesoras - el lado izquierdo coincidente con el instante 0
3. dibujar los bloques correspondientes a las tareas que sólo dependen de las tareas ya introducidas en el diagrama
4. repetir hasta haber dibujado todas las tareas



## AON (Activity on the node)



## AON - características

- ▶ la unidad de análisis básica es la actividad
- ▶ cada actividad es representada por un rectángulo llamado nodo de actividad
- ▶ las flechas representan las relaciones de predecesores y sucesores entre actividades
- ▶ para crear el **Precedence Diagramming Method (PDM)** se necesita determinar las actividades predecesoras y sucesoras de cada tarea
- ▶ nos debemos preguntar: ¿qué actividades deben estar terminadas antes de que pueda comenzar esta actividad?



## AON PDM - construcción

- ▶ el diagrama se hace siguiendo un orden de secuencia lógico, de tal modo que sea leído de izquierda a derecha
- ▶ cada actividad en el diagrama - con excepción de las actividades de comienzo y fin, debe tener:
  - ▶ al menos una actividad que se realice antes (inmediata predecesora)
  - ▶ una actividad que se realice después (su inmediata sucesora)
- ▶ una actividad comienza cuando su predecesora ha sido completada
- ▶ son redes conectadas



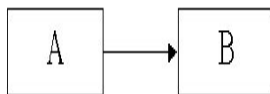
## Dependencias

- ▶ **dependencia:** es una relación que existe entre un par de actividades
- ▶ existen 4 tipos de dependencias:
  - ▶ fin a inicio (FI)
  - ▶ inicio a inicio (II)
  - ▶ inicio a fin (IF)
  - ▶ fin a fin (FF)



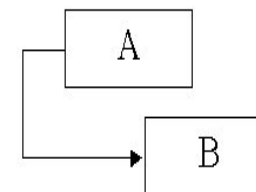
## Dependencia fin a inicio (FI)

- ▶ significa que la actividad A debe estar completada antes que la actividad B comience
- ▶ ejemplo: codificación módulo A y testing módulo A



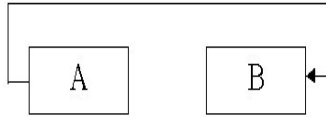
## Dependencia inicio a inicio (II)

- ▶ significa que la actividad B debe comenzar después de que la actividad A ha comenzado
- ▶ no hay relación de tiempo, pueden comenzar juntas o con una diferencia
- ▶ ejemplo: codificación módulo A y generación datos de prueba módulo A



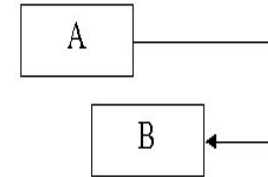
## Dependencia inicio a fin (IF)

- ▶ significa que la actividad B no puede terminar antes que la actividad A haya comenzado
- ▶ ejemplo: no se puede discontinuar el sistema viejo, si el nuevo no comenzó a funcionar



## Dependencia fin a fin (FF)

- ▶ significa que la actividad B no puede terminar antes que termine la actividad A
- ▶ ejemplo: no puede terminar el diseño de módulos hasta que no termine el diseño de las interfaces



## Restricciones

- ▶ el tipo de dependencias entre las actividades es determinado por las restricciones que existen entre esas actividades
- ▶ existen 4 tipos de restricciones:
  - ▶ restricciones técnicas
  - ▶ restricciones gerenciales
  - ▶ restricciones inter-proyectos
  - ▶ restricciones de fechas
- ▶ cada tipo de restricciones puede generar cualquiera de los 4 tipos de dependencias



## Restricciones Técnicas

- ▶ son aquellas que surgen cuando una actividad (la sucesora) requiere un output de otra (la predecesora) para poder comenzar a trabajar
- ▶ en el caso más simple, la predecesora debe estar completada antes que la sucesora pueda comenzar
- ▶ es aconsejable usar dependencias fin a comienzo
- ▶ dentro de esta categoría se pueden presentarse 4 situaciones:
  - ▶ restricciones discrecionales
  - ▶ restricciones por mejores prácticas
  - ▶ restricciones lógicas
  - ▶ requerimientos únicos



## Restricciones Técnicas

- ▶ **restricciones discretionales:** son juicios del LP que resultan en el agregado de una dependencia
- ▶ puede darse como consecuencia de una estrategia de prevención de riesgos
- ▶ **restricciones por mejores prácticas:** son experiencias pasadas que han dado buenos resultados en proyectos anteriores, documentadas en bibliografía y reportes en la industria
- ▶ ejemplo: convertir una dependencia fin a comienzo por una comienzo a comienzo, para acortar tiempos del proyecto



## Restricciones Gerenciales

- ▶ son debido a decisiones políticas de la empresa
- ▶ la diferencia de las restricciones gerenciales con las restricciones técnicas es que las primeras pueden ser revertidas



## Restricciones Técnicas

- ▶ **restricciones lógicas:** son como las restricciones discretionales, surgen de cómo el LP quiere organizar las actividades pero con una justificación razonable
- ▶ **requerimiento único:** sucede cuando existe un recurso crítico necesario para varias actividades
- ▶ para asegurar que no haya conflictos el LP crea dependencias fin a comienzo



## Restricciones Inter-proyectos

- ▶ resultan cuando los entregables de un proyecto son necesarios para otro proyecto
- ▶ las restricciones inter-proyectos surgen en grandes proyectos que se descomponen en sub-proyectos



## Restricciones de Fechas

- ▶ las restricciones de fecha imponen las fechas de comienzo o fin a una actividad de acuerdo a una planificación
- ▶ bajo todo punto de vista deben evitarse - crean complicaciones innecesarias
- ▶ pueden ser de tres tipos:
  - ▶ **no antes que**: especifica la fecha más temprana en la cual una actividad puede completarse. Más leve
  - ▶ **no después de**: especifica la fecha en la cual una actividad debe completarse. Intermedia
  - ▶ **en esta fecha**: especifica la fecha exacta en la cual una actividad debe completarse. Más grave



## Variable Lag

- ▶ las pausas o demoras entre las actividades se muestran en el diagrama mediante el uso de **variable lag**
- ▶ ejemplo: la actividad de registrar datos de una encuesta, comienza 5 días después de enviar las encuestas
- ▶ en este caso se define una dependencia comienzo a comienzo entre las actividades, pero la actividad B comienza 5 días después del comienzo de la actividad A
- ▶ los 5 días se representan mediante una variable lag



## Planificaciones

- ▶ para establecer la planificación del proyecto, se calculan dos planificaciones: la **planificación temprana**, calculada con una estrategia forward, y la **planificación tardía** usando una estrategia backward
- ▶ la planificación temprana consiste del tiempo más temprano que una actividad puede comenzar y terminar
- ▶ la planificación tardía consiste del último tiempo en el cual una actividad puede comenzar y terminar
- ▶ la planificación temprana y tardía son números calculados que se derivan de las dependencias entre las actividades del proyecto



## Planificaciones

- ▶ la combinación de estas planificaciones proveen dos informaciones adicionales:
  - ▶ la **ventana de tiempo** en la cual cada actividad debe comenzarse y terminarse en orden a completar el proyecto en tiempo
  - ▶ la **fecha de terminación del proyecto**
- ▶ **camino crítico**: es la secuencia de actividades que determinan la fecha de terminación del proyecto



## Camino Crítico

- ▶ puede ser definido de varias maneras:
  - ▶ es el camino de duración más larga en el diagrama de red
  - ▶ es la secuencia de actividades cuya planificación temprana y planificación tardía es la misma
  - ▶ es la secuencia de actividades con margen cero (margen = slack)
- ▶ **margen**: es la demora expresada en unidades de tiempo, de la fecha de comienzo o fin de una actividad, que puede ser tolerada sin causar demoras en la finalización del proyecto



## Camino Crítico

- ▶ **actividades del Camino Crítico**: son aquellas actividades que definen al camino crítico
- ▶ cualquier demora en una actividad del camino crítico, demorará la terminación del proyecto con la demora ocurrida en esa actividad



## Comienzo más temprano

- ▶ **comienzo mas temprano (CMTe)** de una actividad (CMTe) es el tiempo más temprano en el cual todas las actividades predecesoras han sido terminadas y la actividad en cuestión puede comenzar
- ▶ El CMTe de una actividad sin predecesoras se fija arbitrariamente en 1, el primer día en que se comienza a trabajar en el proyecto
- ▶ el CMTe de una actividad con mas de una predecesora está dado por el máximo tiempo de finalización más temprano de las actividades predecesoras, más una unidad de tiempo

$$CMTe(p) = \max_{\text{predesor } q \text{ de } p} \{FMTe(q)\} + 1$$



## Finalización más temprana

- ▶ la **finalización mas temprana (FMTe)** de una actividad es calculada como el comienzo más temprano de la actividad más la duración menos uno

$$FMTe(p) = CMTe(p) + duracion(p) - 1$$

- ▶ ejemplo: duración de la actividad: 20 días, comienzo mas temprano: día 45 la finalización mas temprana de la actividad:  $45 + 20 - 1 = 64$
- ▶ la actividad puede finalizar el día 64 como la fecha mas temprana





## Finalización más tardía

- ▶ la **finalización mas tardía (FMTa)** de una actividad es el tiempo más tarde que puede finalizar una actividad sin causar demoras en la finalización del proyecto
- ▶ el tiempo de terminación más tardío de una actividad es determinado por el mínimo de los comienzos más tardíos menos una unidad de tiempo, de todas las actividades sucesoras

$$FMTa(p) = \min_{\text{sucesor } q \text{ de } p} \{ CMTa(q) \} - 1$$

- ▶ si no tiene sucesoras, entonces coincide con el FMTe de la misma tarea (para optimizar la duración total del proyecto)



## Comienzo más tardío

- ▶ el **comienzo mas tardío (CMTa)** de una actividad es el tiempo más tarde que puede comenzar una actividad sin causar demoras en la finalización del proyecto
- ▶ CMTa es calculado como la finalización más tardía menos duración más una unidad de tiempo

$$CMTa(p) = FMTa(p) - \text{duracion}(p) + 1$$

- ▶ ejemplo: duración de la actividad: 20 días, finalización mas tardía: día 90 el comienzo mas tardío de la actividad:  $90 - 20 + 1 = 71$

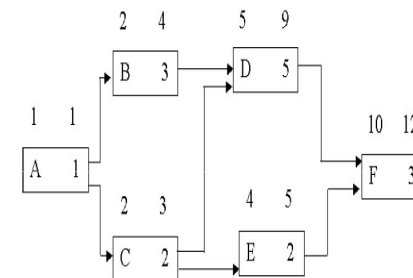


## Ventana de tiempo

- ▶ la **ventana de tiempo** entre el comienzo más temprano y la finalización más tardía, es el período en el cuál se pueden planificar los recursos sin demorar la terminación del proyecto
- ▶ para calcular los tiempos más tardíos y la ventana de tiempo se trabaja en el diagrama de atrás hacia delante



## Ejemplo - planificación más temprana

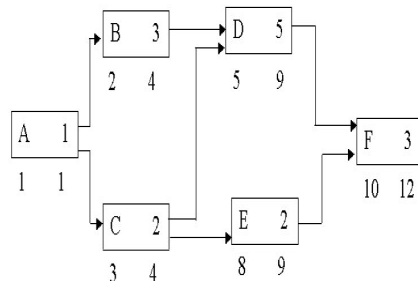


## Cálculo del Camino Crítico

- ▶ una forma de calcular el camino crítico es identificar todos los caminos posibles a través del diagrama y sumar las duraciones de las actividades que residen en el camino
- ▶ el camino con la duración más larga es el camino crítico
- ▶ en proyectos grandes esta opción no es factible y trabajamos con un segundo procedimiento basado en márgenes (el margen de una actividad)



## Ejemplo - planificación más tardía



- ▶ ejemplo: consideremos la actividad C,  $CMTe(C) = 2$ ,  $FMTa(C) = 4$ , duración: 2
- ▶ ventana: 3 días - se debe terminar sin afectar el comienzo más temprano de sus actividades sucesoras (actividades D y E)
- ▶ por lo tanto margen libre: 1 día



## Margen de una actividad

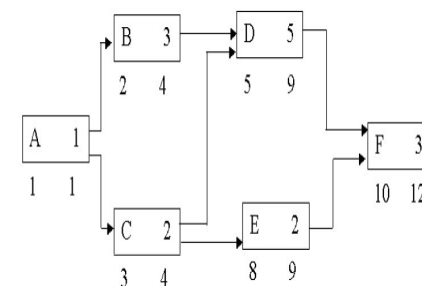
- ▶ es la demora expresada en unidades de tiempo, de la fecha de comienzo o fin de una actividad, que puede ser tolerada sin causar demoras en la finalización del proyecto

$$\text{margen}(p) = FMTa(p) - FMTe(p)$$

- ▶ existen dos tipos de márgenes:
  - ▶ **margen total**: es el rango de fechas en el cual una actividad puede terminar sin demorar la fecha de terminación del proyecto
  - ▶ **margen libre**: es el rango de fechas en el cual una actividad puede terminar sin causar demoras en la planificación más temprana de sus inmediatas sucesoras



## Ejemplo



- ▶ ejemplo:  $CMTe(E) = 4$ ,  $FMTa(E) = 9$ , duración: 2
- ▶ ventana: 6 días - se debe terminar sin afectar el comienzo más temprano de sus actividades sucesoras (actividad F)
- ▶ por lo tanto margen libre: 4 días, margen total: 4 días



## Cálculo camino crítico

- ▶ si una actividad tiene margen cero significa que su culminación determina la fecha de terminación del proyecto
- ▶ la secuencia de actividades con margen cero determinan el camino crítico
- ▶ todas las actividades del camino crítico deben realizarse en su planificación más temprana o de lo contrario se atrasará el proyecto



## Análisis del Modelo Inicial

- ▶ generalmente los cálculos iniciales resultan en una fecha de finalización fuera de los fechas requeridas para terminar el proyecto
- ▶ se debe estudiar la forma de reducir la duración total del proyecto
- ▶ se estudia el modelo para identificar áreas donde se pueda comprimir la duración del proyecto
- ▶ se buscan pares de actividades que actualmente se desarrollan en secuencia y que se puedan ejecutar en paralelo
- ▶ el trabajo de una actividad sucesora puede comenzar una vez que se haya alcanzado cierto grado de avance con la actividad previa



## Análisis del Modelo Inicial

- ▶ se pueden hacer disponibles los entregables de la actividad predecesora para que la sucesora pueda comenzar
- ▶ esto puede aumentar los riesgos del proyecto debido a que puede ocurrir que se deba rehacer trabajo
- ▶ la compresión de la planificación afecta el marco de tiempo en el cual el trabajo debe realizarse, pero no afecta la cantidad de trabajo a realizar
- ▶ los resultados demandan mayor coordinación y mayor comunicación, especialmente en las tareas que se han afectado por los cambios de dependencias



## Análisis del Modelo Inicial

- ▶ se busca en actividades del camino crítico
- ▶ generalmente se focaliza en las primeras actividades. Esto no es aconsejable, ya que en esta etapa el grupo comienza a funcionar
- ▶ otra alternativa es buscar actividades que sean particionables
- ▶ una vez detectadas las actividades particionables, se debe evaluar como comprimir la planificación comenzado antes la actividad sucesora
- ▶ si se cambia la secuencia de actividades, podría modificarse el camino crítico. Esto sucede si se reduce la duración del camino crítico

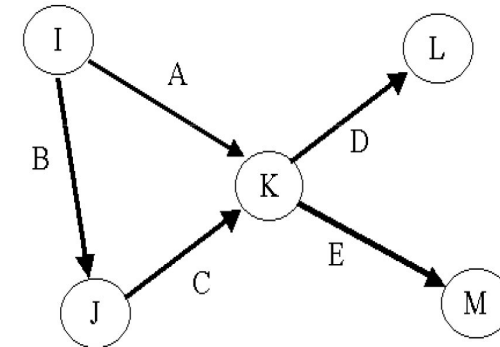


## Reserva Gerencial

- ▶ se agrega una actividad como última actividad del proyecto llamada **reserva gerencial**
- ▶ dependiendo de las características del proyecto puede ser entre un 5% y un 10% de la duración total del mismo
- ▶ se debe administrar el proyecto de manera de no necesitarla
- ▶ sirve como una motivación para el equipo de trabajo. Se puede acortar como sea necesario
- ▶ es útil cuando existen contratos rígidos de cumplimiento de fechas.



## AOA (Activity on the arrow)



## AOA - características

- ▶ cada flecha representa una actividad
- ▶ el nodo a la izquierda de la flecha es el evento que marca el “comienzo de la actividad”
- ▶ el nodo a la derecha de la flecha es el evento que marca el “fin de la actividad”
- ▶ inicialmente los nodos eran numerados secuencialmente y se debía preservar el orden secuencial
- ▶ se pueden agregar actividades fantasmas para preservar la integridad de la red
- ▶ no es muy usado en la actualidad

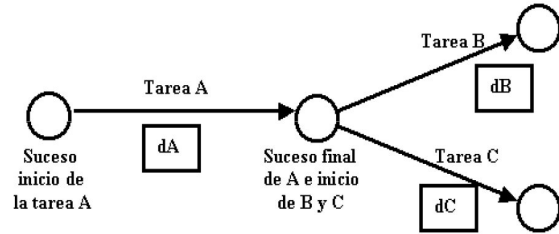


## Método PERT

- ▶ **PERT**: Program Evaluation and Review Technique
- ▶ hipótesis:
  - ▶ un proyecto se divide en tareas
  - ▶ una tarea tiene: nombre, duración, tareas previas
- ▶ características:
  - ▶ **grafo orientado**: conjunto de vértices y vectores
  - ▶ los vectores representan las tareas (AOA)
  - ▶ los vértices representan los sucesos que marcan el inicio y el final de las tareas



## PERT: ejemplo



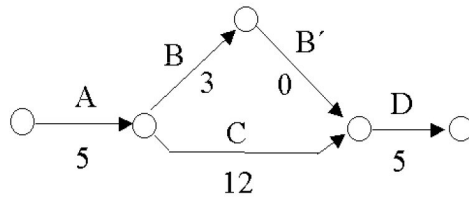
## PERT: condiciones de representación

- ▶ si es necesario, se introduce una **actividad final** (sucesora de todas las actividades sin sucesores) y una **actividad inicial** (predecesoras de todas las actividades sin predecesores)
  - ▶ entre dos vértices, no puede haber más de un vector
- | Tarea | Duración | Tareas Anteriores |
|-------|----------|-------------------|
| A     | 5        | -                 |
| B     | 3        | A                 |
| C     | 12       | A                 |
| D     | 5        | B,C               |
- ▶ si se representara tal cual, no se podrían distinguir los sucesos finales de B y C



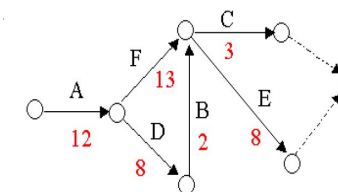
## PERT: condiciones de representación

- ▶ solución: se introduce una tarea ficticia de duración nula



## PERT: ejemplo

Tarea	Duración	Tareas Anteriores
A	12	-
B	2	D
C	3	B,F
D	8	A
E	8	D,B,F
F	13	A



## PERT: valoración de los nodos

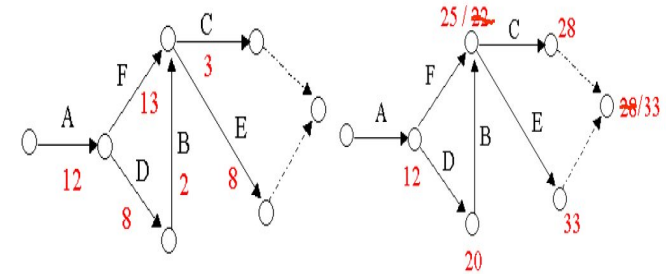
- ▶ se buscan las tareas no condicionadas y se les adjudica valor cero al vértice inicial
- ▶ se valoran los otros vértices examinando todos los valores que llegan al mismo:

$$\text{valorVertice} = \text{valorVerticePrecedente} + \text{duracionTarea}$$

- ▶ se toma el mayor valor para el vértice



## PERT: ejemplo



## PERT: camino crítico

- ▶ **camino crítico**: es el camino que permite alcanzar el valor del último vértice
- ▶ en el ejemplo: el camino crítico está compuesto por las tareas A, F y E



## Modelo COCOMO

- ▶ **COCOMO** Constructive Cost Model
- ▶ desarrollado en la década del '70 por Boehm
- ▶ revisado con una nueva release en 1995 y en el 2000
- ▶ es una colección de tres modelos:
  - ▶ **básico**: aplicable cuando se conoce muy poco del proyecto
  - ▶ **intermedio**: aplicable luego de la especificación de requerimientos
  - ▶ **avanzado**: aplicable cuando se termina el diseño



## Cálculo del Esfuerzo

- ▶ el esfuerzo se calcula mediante la fórmula:

$$PM_{inicial} = c * KLOC^k$$

donde:

- ▶  $PM_{inicial}$ : esfuerzo en personas mes
- ▶  $c$  y  $k$  constantes dadas por el modelo.  $k > 1$
- ▶ se puede corregir mediante [conductores de costos](#)



## Conductores de Costos

- ▶ **atributos del producto**: confiabilidad, complejidad...
- ▶ **atributos computacionales**: restricciones de tiempo de ejecución, de almacenamiento...
- ▶ **atributos del personal**: existe personal experimentado ...
- ▶ **atributos del proceso**: se utilizan herramientas de software sofisticadas...



## Conductores de Costo del COCOMO original

- ▶ **atributos del producto**
  - ▶ confiabilidad requerida
  - ▶ tamaño de la base de datos
  - ▶ complejidad del producto
- ▶ **atributos computacionales**
  - ▶ restricciones tiempo de ejecución
  - ▶ restricciones de almacenamiento
  - ▶ volatilidad de la máquina virtual
  - ▶ tiempo de optimización
  - ▶ experiencia en la máquina virtual



## Conductores de Costo del COCOMO original

- ▶ **atributos del proceso**
  - ▶ uso de prácticas de programación modernas
  - ▶ uso de herramientas de software
  - ▶ complejidad del producto
  - ▶ planificación requerida
- ▶ **atributos del personal**
  - ▶ capacidad de análisis
  - ▶ experiencia en la aplicación
  - ▶ capacidad de programación
  - ▶ experiencia en lenguaje de programación



## COCOMO: pasos básicos

- ▶ los pasos generales son:
  1. estimar el tamaño del software y usar la fórmula del modelo para estimar esfuerzo inicial
  2. revisar la estimación usando conductores de costos u otro factor dado por el modelo
  3. aplicar las herramientas del modelo a la estimación del paso 2 para determinar el total del esfuerzo
- ▶ no es sabio confiar ciegamente en los resultados del modelo
- ▶ es menos sabio ignorar el valor de las herramientas que complementan el juicio experto y la intuición



## COCOMO: cálculo del esfuerzo

- ▶ todos utilizan la misma fórmula:

$$E = a * S^b * F$$

donde:

- ▶  $E$ : esfuerzo en personas mes
- ▶  $S$ : tamaño medido en KDSI (K-delivered source instructions)
- ▶  $F$ : Factor de ajuste (igual a 1 en el modelo básico)
- ▶  $a, b$ : se obtienen de tablas del modelo en función del tipo de sistema



## COCOMO: clasificación de Sistemas

- ▶ clasifica los sistemas en:
  1. **orgánicos**: involucra procesamiento de datos, uso de bases de datos y se focaliza en transacciones y recuperación de datos. Ejemplo: sistema de facturación
  2. **embebidos**: contiene software de tiempo real que es una parte integral de un sistema mayor basado en hardware. Ejemplo: control de ascensores
  3. **semi-embebidos**: entre orgánico y embebido – presenta mayor procesamiento de transacciones. Ejemplo: Monitoreo de una red



## COCOMO: aplicación modelo básico

- ▶ el **modelo básico** aplica las siguientes fórmulas y valores:

$$\text{Personas Mes (PM)} PM = a * (KDSI^b)$$

$$\text{Tiempo de Desarrollo (TD)} TD = c * (PM^d)$$

con los siguientes valores de  $a$  y  $b$ :

tipo	$a$	$b$	$c$	$d$
orgánico	2.40	1.05	2.50	0.38
embebido	3.00	1.12	2.50	0.35
semi-embebido	3.60	1.20	2.50	0.32





## COCOMO: aplicación modelo intermedio

- ▶ cuando se conoce muy poco del proyecto, se utiliza COCOMO básico con  $F = 1$
- ▶ cuando se conoce un poco mas: el lenguaje, herramientas a utilizar se puede aplicar COCOMO intermedio
- ▶ se eligen los **conductores de costos** de una tabla que presenta 15
- ▶ la importancia de cada conductor de costo es clasificada en una escala ordinal con seis puntos: **Muy Baja**, **Baja**, **Nominal**, **Alta**, **Muy Alta**, **Extra Alta**



## COCOMO: conductores de costo

- ▶ se denominan también **atributos de costos**
- ▶ tratan de capturar el impacto del entorno del proyecto en el costo de desarrollo
- ▶ de un análisis estadístico de más de 100 factores que influyen en el costo, Boehm retuvo 15 de ellos para COCOMO
- ▶ se agrupan en cuatro categorías:
  - ▶ **atributos del producto**
  - ▶ **atributos del hardware**
  - ▶ **atributos del personal**
  - ▶ **atributos del proyecto**



## COCOMO: atributos del producto

1. **RELY**: garantía de funcionamiento requerida al software
2. **DATA**: tamaño de la base de datos
3. **CPLX**: complejidad del producto



## COCOMO: atributo RELY

- ▶ indica las posibles consecuencias para el usuario en el caso que todavía existan defectos en el producto
- ▶ **muy baja**: el efecto de un fallo del software simplemente trae como consecuencia la inconveniencia de corregir el fallo
- ▶ **baja**: el efecto de un fallo del software es una pérdida fácilmente recuperable para los usuarios
- ▶ **nominal**: el efecto es una moderada pérdida para los usuarios, pero es una situación de la que se puede salir sin excesiva dificultad
- ▶ **alta**: el efecto es una gran pérdida financiera o una inconveniencia masiva humana
- ▶ **muy alta**: el efecto es una pérdida de vidas humanas

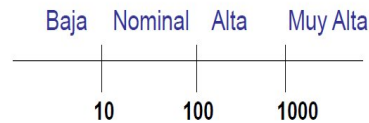


## COCOMO: atributo DATA

- ▶ indica el tamaño de la base de datos a desarrollar en relación con el tamaño del programa
- ▶ existen cuatro segmentos con la razón 10-100-1000, que determinan las puntuaciones de 'bajo' a 'muy alto'
- ▶ se define por el cociente

$$D/P$$

donde  $D$  es el tamaño de la base de datos en bytes y  $P$  es el tamaño del programa en DSI



## COCOMO: atributo CPLX

- ▶ indica la complejidad de cada módulo y se utiliza para determinar la complejidad compuesta del sistema
- ▶ la puntuación puede variar de **muy baja** si el módulo está compuesto de expresiones matemáticas simples a **extra alta** para módulos que utilizan muchos recursos de planificación



## COCOMO: atributos del hardware

1. **TIME**: limitaciones en el porcentaje del uso de la CPU
2. **STOR**: limitaciones en el porcentaje del uso de la memoria
3. **VIRT**: volatilidad de la máquina virtual
4. **TURN**: frecuencia de cambio en el modelo de explotación



## COCOMO: atributo TIME

- ▶ se expresa en el porcentaje de tiempo de ejecución disponible
- ▶ se basa en la presunción de que siempre será más exigente para un programador escribir un programa que tiene una restricción en el tiempo de ejecución
- ▶ es **nominal** cuando el porcentaje es el 50%, y **extra alto** cuando la restricción es del 95%



## COCOMO: atributo STOR

- ▶ captura el esfuerzo de programación para que el programa pueda correr en un volumen menor de almacenamiento principal
- ▶ se basa en la presunción de el esfuerzo de programación se incrementa si el programa tiene que correr en un volumen menor del almacenamiento principal
- ▶ captura este esfuerzo extra desde **nominal** cuando la reducción del almacenamiento principal es del 50% a **extra alta** cuando la reducción es del 95%



## COCOMO: atributo VIRT

- ▶ refleja los cambios que puede sufrir la máquina virtual (hardware más software) durante el desarrollo del software
- ▶ es la probabilidad de que ocurran los cambios desde **baja** a **muy alta**



## COCOMO: atributo TURN

- ▶ cuantifica el tiempo de respuesta del ordenador desde el punto de vista del programador. Cuanto mayor sea el tiempo de respuesta, más alto será el esfuerzo humano
- ▶ puede variar desde **baja** para un sistema interactivo; a **muy alta**, cuando el tiempo medio de respuesta es de más de 12 horas



## COCOMO: atributos del personal

1. **ACAP**: calificación de los analistas
2. **AEXP**: experiencia del personal en aplicaciones similares
3. **PCAP**: calificación de los programadores
4. **VEXP**: experiencia del personal en la máquina virtual
5. **LEXP**: experiencia en el lenguaje de programación a usar



## COCOMO: atributo ACAP

- ▶ mide la capacidad del grupo de analistas, en términos de habilidad de análisis, eficiencia y capacidad para cooperar
- ▶ estas habilidades tienen un impacto significativo en el esfuerzo humano
- ▶ cuanto más capaz sea el grupo, menos esfuerzo será necesario
- ▶ puede variar desde **muy bajo** a **muy alto**



## COCOMO: atributo AEXP

- ▶ mide la experiencia del grupo en una aplicación similar
- ▶ la experiencia del grupo tiene una gran influencia en el esfuerzo
- ▶ **muy bajo**:  $\leq 4$  meses experiencia media
- ▶ **bajo**: 1 año de experiencia media
- ▶ **nominal**: 3 años de experiencia media
- ▶ **alto**: 6 años de experiencia media
- ▶ **muy alto**:  $\geq 12$  años, o reimplementación de un subsistema



## COCOMO: atributo PCAP

- ▶ mide la capacidad del grupo de programadores, en términos de habilidad de programación, eficiencia y capacidad para cooperar
- ▶ similar al atributo que mide la calificación de analistas, pero en este caso, se mide al grupo de programadores
- ▶ cuanto más capaz sea el grupo, menos esfuerzo será necesario
- ▶ se aplica a los programadores como grupo, pero no a los programadores individuales
- ▶ puede variar desde **muy bajo** a **muy alto**



## COCOMO: atributo VEXP

- ▶ mide la experiencia de los programadores en la máquina virtual
- ▶ **muy bajo**:  $< 1$  mes experiencia media
- ▶ **bajo**: 4 meses
- ▶ **nominal**: 1 año
- ▶ **alto**:  $> 3$  años



## COCOMO: atributo LEXP

- ▶ mide la experiencia de los programadores en el lenguaje
- ▶ un grupo de programadores con amplia experiencia en un lenguaje determinado programará de una manera mucho más segura, generando un menor número de defectos y de requerimientos humanos
- ▶ puede variar desde **muy bajo** a **alto** para un grupo de un mes a tres años de experiencia, respectivamente
- ▶ **muy bajo**: < 1 mes experiencia media
- ▶ **bajo**: 4 meses experiencia media
- ▶ **nominal**: 1 año experiencia media
- ▶ **alta**: > 3 años



## COCOMO: atributos del proyecto

1. **MODP**: uso de prácticas modernas de programación
2. **TOOL**: uso de herramientas de desarrollo de software
3. **SCED**: limitaciones en el cumplimiento de la planificación



## COCOMO: atributo MODP

- ▶ indica la utilización de prácticas modernas de programación (PMP)
- ▶ **muy bajo**: no se utilizan PMP
- ▶ **bajo**: uso experimental de algunas PMP
- ▶ **nominal**: experiencia razonable en el uso de algunas PMP en el proyecto
- ▶ **alto**: experiencia razonable en gran parte de las PMP para el proyecto, y amplio uso de las mismas
- ▶ **extra alto**: uso habitual de PMP en el proyecto



## COCOMO: atributo TOOL

- ▶ indica el uso de herramientas de software
- ▶ el uso adecuado de herramientas de software es un multiplicador de la productividad
- ▶ la puntuación de TOOL varía desde **muy bajo** cuando sólo se utilizan herramientas básicas, a **muy alto** cuando se utilizan herramientas específicas



## COCOMO: atributo SCED

- ▶ indica el esfuerzo necesario para cumplir con la planificación
- ▶ el tiempo nominal de desarrollo, tal como se define en el modo básico, es el plazo que requiere menor esfuerzo humano
- ▶ cualquier apresuramiento (**muy bajo**) o retraso (**muy alto**) demandarán más esfuerzo



## COCOMO: índices

Grado	Muy Baja	Baja	Nominal	Alta	Muy Alta	Extra Alta
Atributos del Producto						
RELY	0.75	0.88	1.00	1.15	1.40	
DATA		0.94	1.00	1.08	1.16	
CPLX	0.70	0.85	1.00	1.15	1.30	1.65
Atributos del Hardware						
TIME			1.00	1.11	1.30	1.62
STOR			1.00	1.06	1.21	1.56
VIRT		0.87	1.00	1.15	1.30	
TURN		0.87	1.00	1.07	1.15	



## COCOMO: índices

Grado	Muy Baja	Baja	Nominal	Alta	Muy Alta	Extra Alta
Atributos del Personal						
ACAP	1.46	1.19	1.00	0.86	0.71	
AEXP	1.29	1.13	1.00	0.91	0.82	
PCAP	1.42	1.17	1.00	0.86	0.70	
VEXP	1.21	1.10	1.00	0.90		
LEXP	1.14	1.07	1.00	0.95		
Atributos del Proyecto						
MODP	1.24	1.10	1.00	0.91	0.82	
TOOL	1.24	1.10	1.00	0.91	0.83	
SCED	1.23	1.08	1.00	1.04	1.10	



## COCOMO: modelo avanzado

- ▶ para los proyectos medios, COCOMO intermedio es coincidente con COCOMO básicos
- ▶ el modelo debe ser calibrado al propio entorno de desarrollo
- ▶ una vez que se han identificado los módulos del sistema, se puede utilizar **COCOMO avanzado** o detallado
- ▶ en el modelo COCOMO avanzado se aplica la versión intermedia a nivel de componentes y luego se construye una estimación para el proyecto completo



## COCOMO: fases de desarrollo

- ▶ permite estimar los tiempos para cada una de las fases de desarrollo
- ▶ considera cuatro fases:
  - ▶ requerimientos
  - ▶ diseño del producto
  - ▶ programación
  - ▶ prueba/integración



## COCOMO: fase de diseño

- ▶ el diseño del Producto es la segunda fase del ciclo de desarrollo
- ▶ COCOMO se preocupa de la determinación de la arquitectura del producto y de las especificaciones de los subsistemas
- ▶ esta fase requiere del 16% al 18% del esfuerzo nominal PM
- ▶ puede durar del 19% al 38% del tiempo nominal de desarrollo TD



## COCOMO: fase de requerimientos

- ▶ es la primera fase del ciclo de desarrollo
- ▶ se analizan los requerimientos, se muestra un Plan de Producto y se genera una especificación completa del producto
- ▶ esta fase consume del 6% al 8% del esfuerzo nominal PM
- ▶ puede durar del 10% al 40% del tiempo nominal de desarrollo TD
- ▶ estos porcentajes dependen del modo y del tamaño (de 2000 LOC a 512000 LOC)



## COCOMO: fase de programación

- ▶ la tercera fase del ciclo de desarrollo se subdivide en dos subfases: diseño detallado y prueba del código
- ▶ esta fase requiere del 48% al 68% del esfuerzo nominal PM
- ▶ puede durar del 24% al 64% del tiempo nominal de desarrollo TD



## COCOMO: fase de prueba/integración

- ▶ esta fase consiste principalmente en unir las diferentes unidades ya probadas
- ▶ se utiliza del 16% al 34% del esfuerzo nominal PM
- ▶ puede durar del 18% al 34% del tiempo nominal de desarrollo TD



## COCOMO 2.0: modelos

1. **Application Composition Model** orientada a apoyar la estimación en las etapas tempranas del desarrollo de software, donde básicamente se cuenta con escasa información sobre los requerimientos del sistema. Este modelo además apoya la estimación para el desarrollo de prototipos rápidos en cualquier etapa del ciclo de vida del software. Utiliza **puntos objeto**
2. **Early Design Model** enfocada a apoyar la estimación tras la definición de los requerimientos y así respaldar la toma de decisiones sobre la evaluación de arquitecturas y conceptos de operación alternativos para el desarrollo de software. Utiliza **puntos función**
3. **Post-Architecture Model** orientada a apoyar la estimación durante el desarrollo del sistema. Más detallado, con nuevos conductores de costo y ecuaciones. Utiliza **KDSI**.



## COCOMO 2.0

- ▶ el modelo original de COCOMO resultó muy exitoso, sin embargo su aplicación no es práctica para entornos modernos de desarrollo
- ▶ es una actualización de COCOMO para que se adapte a las nuevas tecnologías, enfoques de OO, etc. Fue publicado en el año 2000
- ▶ objetivos
  - ▶ desarrollar modelos de costos y de estimación acordes a las prácticas actuales
  - ▶ desarrollar bases de datos de costos y herramientas que soporten una mejora continua del modelo
  - ▶ proveer un framework analítico cuantitativo, y un conjunto de herramientas y técnicas para evaluar los efectos de las mejoras en los costos de ciclos de vida y en las planificaciones



## COCOMO 2.0: cálculo puntos objetos

1. estimar el número de pantallas, reportes y componentes 3GL
2. clasificar cada uno de estos objetos en complejidad: **simple**, **media**, o **alta**, si son pantallas

Número de vistas contenidas	Número y Fuente de las Tablas de Datos		
	Total < 4 ( < 2 server, < 3 cliente)	Total < 8 ( 2-3 server, 3-5 cliente)	Total +8 ( > 3 server, > 5 cliente)
< 3	Simple	Simple	Mediana
3 - 7	Simple	Mediana	Alta
+ 8	Mediana	Alta	Alta





## COCOMO 2.0: cálculo puntos objetos

1. estimar el número de pantallas, reportes y componentes 3GL
2. clasificar cada uno de estos objetos en complejidad: **simple**, **media**, o **alta**, si son reportes

Número de secciones contenidas	Número y Fuente de las Tablas de Datos		
	Total < 4 ( < 2 server, < 3 cliente)	Total < 8 ( 2-3 server, 3-5 cliente)	Total +8 ( > 3 server, > 5 cliente)
0 - 1	Simple	Simple	Mediana
2 - 3	Simple	Mediana	Alta
+ 4	Mediana	Alta	Alta



## COCOMO 2.0: cálculo puntos objetos

3. pesar los objetos de acuerdo a su complejidad en base a la siguiente tabla:

Tipo de Objeto	Peso en base a Complejidad		
	Simple	Media	Alta
Pantalla	1	2	3
Informe	2	5	8
Componente 3GL			10



## COCOMO 2.0: cálculo puntos objetos

4. determinar los **Puntos Objetos (PO)**, sumando todos los pesos de las instancias
5. estimar el porcentaje de reuso que se espera y calcular los **Nuevos Puntos Objetos (NPO)**

$$NPO = PO(100 - \%reuso)/100$$



## COCOMO 2.0: cálculo puntos objetos

6. determinar el ratio de productividad, en base a:

$$PROD = NPO / personasMes$$

Experiencia y capacidad de desarrolladores	Muy baja	Baja	Nominal	Alta	Muy alta
Madurez y capacidad del ICASE	Muy baja	Baja	Nominal	Alta	Muy alta
PROD	4	7	13	25	50



## COCOMO 2.0: cálculo puntos función

1. calcular los puntos función, en base a requerimientos y documentos de diseño. Para calcularlos, contar:
  - 1.1 inputs externos (IE)
  - 1.2 outputs externos (OE)
  - 1.3 consultas externas (CE)
  - 1.4 archivos internos (AI)
  - 1.5 archivos de interface externos (AE)



## COCOMO 2.0: cálculo puntos función

2. clasificar cada punto función de acuerdo a su complejidad de acuerdo a las siguientes tablas:

Para AI y AE				Para OE y CE			
Registros	Datos			Tipos de Archivos	Datos		
	1-19	20-50	+ 51		1-5	6-19	+ 20
1	Baja	Baja	Media	0 - 1	Baja	Baja	Media
2 - 5	Baja	Media	Alta	2 - 3	Baja	Media	Alta
6 - +	Media	Alta	Alta	4 - +	Media	Alta	Alta

Para IE			
Tipos de Archivos	Datos		
	1-4	5-15	+ 16
0 - 1	Baja	Baja	Media
2 - 3	Baja	Media	Alta
4 - +	Media	Alta	Alta



## COCOMO 2.0: cálculo puntos función

3. aplicar pesos de complejidad, de acuerdo a:

Puntos Función	Peso de Complejidad		
	Baja	Media	Alta
Inputs externos	3	4	6
Outputs externos	4	5	7
Consultas externas	3	4	6
Archivos internos	7	10	15
Archivos externos	5	7	10

4. calcular los **puntos función no ajustados (PFNA)**, sumando los puntos de función por complejidad



## COCOMO 2.0: cálculo puntos función

5. convertir puntos de función a líneas de código, en base a las siguientes proporciones:

Lenguaje	SLOC/PFNA	Lenguaje	SLOC/PFNA
Ada	71	Java	53
Basic (compilado)	91	Lisp	64
Basic (interpretado)	128	Modula 2	80
C	128	Pascal	91
C++	29	Prolog	64
ANSI Cobol 85	91	Generador de reportes	80
Fortran 77	105	Planilla de cálculo	6



## COCOMO 2.0: cálculo esfuerzo

- ▶ en COCOMO II el esfuerzo es expresado en Personas Mes (PM)

$$PM_{nominal} = a * (Tamaño)^b$$

el tamaño es expresado en KSLOC

- ▶ *a* intenta cuantificar los efectos multiplicativos en el esfuerzo de proyectos de tamaño creciente
- ▶ *b* intenta medir la economía (o no economía) de escala encontrada en proyectos de diferentes tamaños
- ▶ si  $b < 1,0$  el proyecto exhibe economía de escala (la productividad aumenta a medida que aumenta el tamaño del producto)
- ▶ si  $b = 1,0$  la economía, o no economía están balanceadas
- ▶ si  $b > 1,0$  el proyecto no exhibe economía de escala (aumento de comunicación, problemas de integración)



## Otros modelos

- ▶ SLIM (Putnam)
- ▶ DBA COCOMO 2004
- ▶ COINCOMO 2004
- ▶ COCOTS 2000
- ▶ COSYSMO 2002
- ▶ COSoSIMO 2004
- ▶ Costing Secure System 2004



## Problemas con los modelos existentes

- ▶ hay acuerdo en que el **tamaño del producto** es factor determinante del esfuerzo requerido para construir el producto
- ▶ la mayoría de los modelos sugiere que el esfuerzo requerido es aproximadamente proporcional al tamaño
- ▶ incluyen un ajuste (*b*...) inverso a la economía de escala, proyectos mas grandes son menos productivos que los mas pequeños
- ▶ el modelo de Putnam concluye que disminuyendo la duración aumenta el esfuerzo, pero aumentando la duración disminuye el esfuerzo



## Problemas con los modelos existentes

- ▶ los modelos pueden trabajar bien para los entornos en los cuales fueron derivados, no parece razonable que un modelo simple pueda ser válido universalmente
- ▶ los modelos son muy sensible al factor tecnológico, que a su vez no es fácil de obtener
- ▶ la mayoría de los modelos trabajan sobre un conjunto de datos de análisis post hoc
- ▶ los parámetros de input necesarios son muy difíciles de obtener en las primeras etapas del proyecto



## Mejoras de las Predicciones

- ▶ **definición de datos locales:** usar medidas de tamaño y esfuerzo que son definidas consistentemente a través de todo el entorno
- ▶ **proceso de calibración:**
  - ▶ asegurar que los valores provistos al modelo son consistentes con los requerimientos y expectativas del modelo
  - ▶ reajustar los coeficientes del modelo usando datos de proyectos pasados para reflejar la productividad básica encontrada en el nuevo entorno
  - ▶ eliminar, agregar o modificar conductores de costos
  - ▶ personalizar el modelo



## Mejora de las Predicciones

- ▶ **grupo de estimación independiente:** propuesta de De Marco. Asegura consistencia entre proyectos, mayor experiencia, monitoreo de la base de datos
- ▶ **reducir subjetividad al input:** los modelos de estimación locales basados en datos mas homogéneos son mas exactos que modelos mas generales y complejos. Estos modelos locales reducen subjetividad
- ▶ **estimación preliminar y re-estimación:** la estimación temprana requiere el uso de información incompleta. Se mejora relacionando dos pasos: estimación preliminar y re-estimación cuando se dispone de la información



## Grupo de Estimación

- ▶ el coordinador le entrega a cada estimador la información relevante
- ▶ el coordinador convoca a una reunión donde se hacen preguntas y se discute aspectos de la estimación
- ▶ cada experto provee una estimación (rango) inicial anónima
- ▶ el coordinador circulariza un resumen de las estimaciones individuales y el promedio
- ▶ el coordinador convoca a otra reunión para discutir los resultados
- ▶ los expertos proveen nuevas estimaciones anónimas
- ▶ se repiten pasos 2 a 6 hasta que el grupo decide que el promedio es representativo y satisfactorio
- ▶ Putnam propone:

$$estimacion = (cotaInferior + 4 * valoresSimilares + cotaSuperior)$$



## Propuesta de Proyecto

- ▶ el entregable de las JPPS es la **propuesta de proyecto**
- ▶ este documento será enviado a los gerentes senior para su aprobación
- ▶ establece la descripción completa de la oportunidad de negocio del proyecto
- ▶ incluye: valor de negocio esperado, costos, cronogramas, planes de riesgos y otros apéndices
- ▶ es de esperar varias revisiones antes de la obtención de la aprobación



## Propuesta de Proyecto: posible estructura

- ▶ **Resumen ejecutivo** muy corto, contiene generalmente la situación del negocio, el objetivo del proyecto y su valor de negocio, contenido expandido del POS
- ▶ **Antecedentes** es una breve descripción de la situación que motivó el proyecto (condiciones, problemas, oportunidades)
- ▶ **Objetivos** una breve descripción de lo que se espera obtener del proyecto
- ▶ **Estrategia** para aquellos que no están interesados en los detalles de cómo conseguir los objetivos, esta es una breve descripción de las tareas (modelo de PMLC) a usar



## Propuesta de Proyecto: posible estructura

- ▶ **Descripción Detallada del Trabajo** es un resumen de alto nivel de lo que se realizará, cuándo y quién lo realizará, cuánto tiempo demandará, y los criterios para medir completitud. Los diagramas de Gantt son útiles en este contexto
- ▶ **Resumen de Tiempos y Costos** una tabla resumen que incluya tiempos y costos de las actividades
- ▶ **Apéndices** incluyen todos los datos de soporte de contenidos anteriores, anticipando preguntas de los revisores y sus respuestas



## Obteniendo la aprobación

- ▶ la propuesta de proyecto debe convencer de que el proyecto es una buena oportunidad de negocio, y que los recursos necesarios están en línea con esta oportunidad
- ▶ las revisiones pueden ocurrir porque
  - ▶ el **costo/beneficio** no está a favor. Se debe descomponer la solución y estimar los beneficios por funciones, posiblemente eliminando funciones con poco beneficio o mucho costo
  - ▶ los **riesgos** de fracaso son altos. Se debe tratar de eliminar los riesgos buscando otras soluciones.
  - ▶ el **costo** excede la posibilidad de financiamiento. Analizar el triángulo de alcance o descomponer el proyecto en fases
  - ▶ otros proyectos **compiten** por los mismos recursos. Quizás es tiempo de presionar los “sponsors” en la política de la organización

