

# CC51A – Ingeniería de Software

## *Introducción a la Gestión de Proyectos de Software*

Sergio Ochoa D.

Parte de este material ha sido cedido por el Prof. Jaime Navón, PhD.

([www.ing.puc.cl/~jnavon](http://www.ing.puc.cl/~jnavon))

# Estructura de la Presentación

1. Principios Básicos
2. Planificación de Proyectos
3. Administración de Proyectos
4. Malas Noticias
5. Síntomas Comunes
6. Deficiencias Comunes
7. Conclusiones.

# Principios Básicos de la Adm. de Proyectos

“Cualquier comandante que acepta llevar a cabo un plan que él considera defectuoso es culpable; él debe exponer sus razones, e insistir en los cambios necesarios y finalmente ofrecer su dimisión antes de ser el instrumento de la derrota de su ejército”.

**Napoleón**

- ◆ Si no estamos convencidos,.... no podremos administrar nada.
- ◆ La administración (o gestión) requiere: *planificación y control*.
  - La *planificación* del proyecto consiste en distribuir la estimación del esfuerzo a lo largo del proyecto asignando.

# Principios Básicos de la Adm. de Proyectos (cont...)

- La *planificación* consiste en establecer los hitos y las actividades para chequear el avance del proyecto.
- La *planificación* del proyecto consiste en redefinir la planificación sobre la marcha. Aportando medidas para palear los desfases respecto a la planificación inicial.
- El *control (o seguimiento)* del proyecto consiste en ejecutar las actividades de revisión previstas en la etapa de planificación, y comparar los resultados contra lo planificado.

*Para hacer esto hay que seguir una serie de Principios Básicos...*

# Principios Básicos de la Adm. de Proyectos (cont...)

- **Segmentación** - el proyecto debe ser separado en un número manejable de actividades y tareas.
- **Interdependencia** – la planificación debe reflejar la segmentación de tareas, y la relaciones entre ellas. Algunas tareas ocurren en secuencia, otras en paralelo, algunas son requisito de otras, etc.
- **Asignación de tiempo** - a cada tarea se le debe asignar un cierto número de unidades de trabajo (personas-días, etc) y fechas de inicio y término.
- **Validación del esfuerzo** – se debe verificar que la gente afectada a una tarea esté disponible, que la cantidad no esté sobredimensionada, y además que sea suficiente.

# Principios Básicos de la Adm. de Proyectos (cont...)

- **Definición de responsabilidades** - cada tarea debe tener un responsable.
- **Salida definida** - cada tarea debe tener una salida o producto bien definido. Por ejemplo: diseño de un módulo, plan de pruebas, etc.
- **Definición de metas** - cada grupo de tareas debe estar asociado con una meta.

# Planificación de Tareas

◆ Se pueden usar cosas como PERT o CPM

◆ Cualquiera de estos métodos provee mecanismos para:

- determinar la ruta crítica (secuencia de tareas que define la duración del proyecto)
- establecer estimaciones probables para tareas individuales
- calcular tiempos límites que definen ventanas de tiempo para las distintas actividades
  - ◆ lo más temprano que una actividad puede empezar si todas las precedentes se completan en el mínimo tiempo
  - ◆ lo más tarde que puede comenzar una tarea sin aumentar el tiempo mínimo del proyecto
- existen herramientas de software para ello

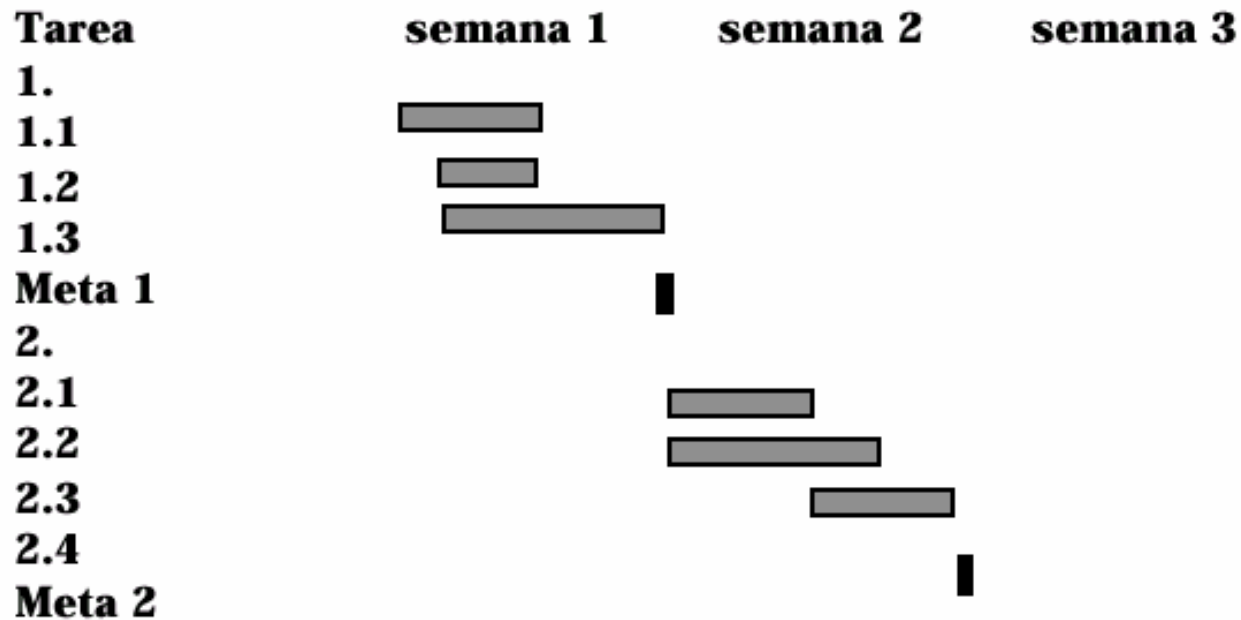


# Planificación de Tareas (cont...)

**Set de tareas**

**Carta Gantt**

**Tablas de  
tareas**





# Control de Tareas

Tareas	Comienzo		Termino		Resp	Esfuerzo
	plan	real	plan	real		
1.						
1.1	wk1d1	wk1d1	wk1d3	wk1d4	JNC	3 pd
1.2	wk1d2	wk1d2	wk1d2	wk1d2	ACU	1 pd
...	...	...	...	...	...	...

- ◆ Hay que contrastar la realidad contra lo planificado, ... y generar planes de contingencia, en caso de ser necesario.
- ◆ Hay que monitorear (controlar) todo, así me aseguraré de tener “las riendas el proyecto”.
- ◆ Monitorear es un medio.... No un fin....

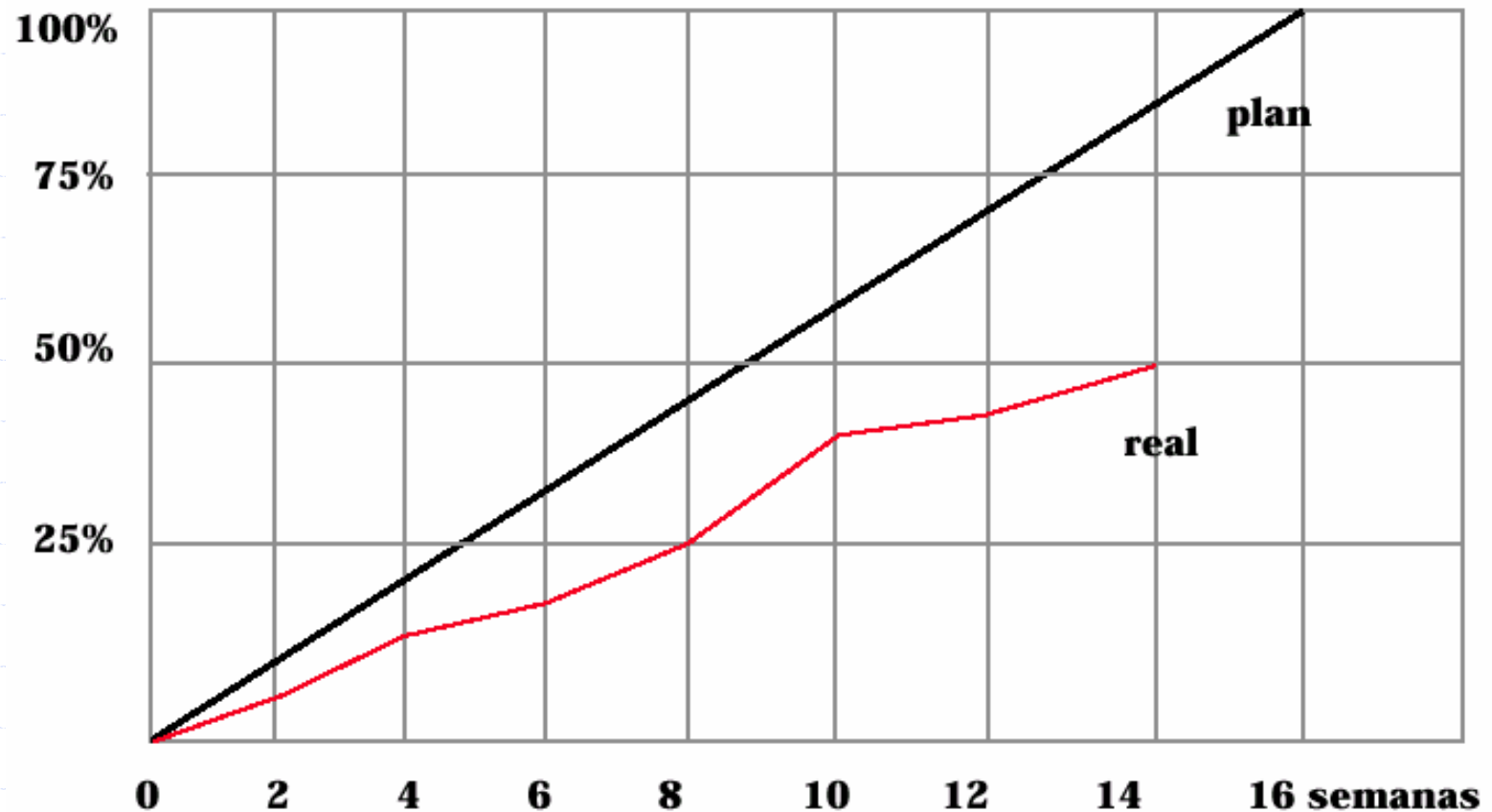
# Control de Tareas (cont...)

## **Formas de monitoreo :**

- llevar a cabo reuniones periódicas en las cuales los participantes informan del estado de las tareas, el progreso o los problemas encontrados
- determinar si las metas están siendo cumplidas de acuerdo a lo programado (checkpoints)
- comparar tiempos planeados contra tiempos reales, para las distintas tareas. Emitir alertas en caso de ser necesario (por ej. Para disparar un plan de contingencia)

# Control de Tareas (cont...)

**El Gráfico de Avance**



# Control de Tareas (cont...)

## Al detectar problemas con la planificación:

- ◆ Asignar más recursos al área afectada
- ◆ Readecuar el personal en general
- ◆ Cambiar la programación del proyecto
- ◆ Usar técnicas de “time-box”
  - Básicamente, es una estrategia de tipo incremental.
  - Se define la entrega del primer incremento, y luego se reajustan todas las tareas de acuerdo a este incremento.
  - Al llegar al límite de tiempo de una tarea para ese incremento, esa tarea se detiene y comienza la siguiente
  - Aunque una tarea no esté completa, muchas veces es posible postergar la última parte para el siguiente incremento

# Máximas....

- “la mayoría de las veces el administrador del proyecto marca la diferencia entre el éxito o el fracaso de un proyecto”
- “la gestión es tan importante como la parte técnica....”
- “... es difícil encontrar un buen administrador de proyectos”
- “La administración de proyectos es una disciplina que se debe aprender....”
- “ En un equipo de trabajo, el adm. de proyectos es quien peor la pasa”

# ¿Cómo Mejorar la Adm. de Proyectos de Software ?

- ◆ Requiere saber ¿dónde estamos hoy?. Con exactitud !!!
- ◆ Requiere revisar/rediseñar:
  - El proceso de administración de proyectos actuales.
  - La ingeniería de requisitos.
  - La planificación de proyectos.
  - El seguimiento y control de proyectos.
  - El aseguramiento de la calidad.
  - La administración de la configuración.

# Malas Noticias

- ◆ La expansión de los sistemas de software en tamaño, complejidad, distribución e importancia empuja los límites de lo que hasta la fecha sabíamos desarrollar.
- ◆ Tratar de avanzar más allá de los sistemas legados, hacia tecnologías más modernas, trae su propio conjunto de problemas técnicos y organizacionales.
- ◆ La administración de proyectos de software se hace cada vez más difícil...



# Malas Noticias (cont...)

- ◆ La competencia obliga a las empresas de desarrollo a mejorar su productividad, manteniendo o mejorando la calidad de sus productos, realizando además desarrollos más rápidos y económicos.
- ◆ Adicionalmente, la velocidad de preparación de personal de desarrollo calificado, es menor que la requerida.
- ◆ Además, actualmente el personal tiene competencias adicionales a las tradicionales (por ej. diseño gráfico, diseño de procesos de negocio, etc).

# Malas Noticias (cont...)

- ◆ La construcción y mantención de software es una tarea difícil y será aún más difícil.
- ◆ Construir software de una cierta calidad, en forma repetible, es una tarea aún más difícil.
- ◆ Todo esto se refleja en la Administración de Proyectos.... Este es el *por qué?* de las máximas antes mencionadas

# Síntomas Comunes

- ◆ Imprecisión en el entendimiento de las necesidades del usuario final.
- ◆ Falta de habilidad para manejar requisitos cambiantes.
- ◆ Módulos que no calzan....
- ◆ Software que es difícil de mantener o extender.

# Síntomas Comunes (cont...)

- ◆ Descubrimiento tardío de fallas serias.
- ◆ Problemas ajenos, como el Y2K.
- ◆ Miembros del equipo en muchas cosas, o ausentes, haciendo imposible reconstruir historia de cambios.
- ◆ Falta (o problemas) de comunicación entre los miembros del equipo.

# Deficiencias Comunes

- ◆ Administración de requisitos ad-hoc.
- ◆ Comunicación ambigua e imprecisa.
- ◆ Responsabilidades y roles difusos.
- ◆ Arquitecturas “escabrosas”.
- ◆ Complejidad muy grande. Falta de segmentación del problema
- ◆ Inconsistencias no detectadas en requisitos, diseño e implementación.
- ◆ Testeo insuficiente.

# Deficiencias Comunes (cont...)

- ◆ Evaluación subjetiva del estado del proyecto.
- ◆ Incapacidad de los miembros para trabajar en grupo.
- ◆ Fallas para analizar y atacar los riesgos.
- ◆ Propagación de cambios no controlada.
- ◆ Automatización insuficiente.

# Buenas Prácticas de Software

- ◆ Para construir buen software no basta con ser buen programador.
- ◆ Para ello, hay que conocer, planificar y controlar los procesos y recursos asignados a un proyecto.
- ◆ Existe una nueva ola en ingeniería de software, basada en dichos principios, que permite visualizar el futuro en forma más optimista.



# Conclusiones

- ◆ La planificación del proyecto provee una definición de cada macro-tarea, junto con estimaciones de tiempos y recursos requeridos.
- ◆ La planificación del proyecto provee además un marco de referencia para monitoreo y control de estas tareas.
- ◆ La planificación del proyecto se desarrolla al comienzo pero es continuamente refinada a medida que el trabajo progresa.
- ◆ Salvo raras excepciones, los recursos y tiempos estimados inicialmente, difícilmente son alcanzados.

# Conclusiones (cont...)

- ◆ Hay que controlar (monitorear) todo... pero el monitoreo es un medio, no un fin.
- ◆ Si el trabajo no coincide con los tiempos y recursos asignados, debe recortarse la tarea, o aumentarse los tiempos y/o recursos asignados.
- ◆ Cita clásica....
  - “si se incorpora más gente a un proyecto que ya está atrasado, éste se atrasará aún más” (Brooks, 1975)
- ◆ Es necesario ir comparando las estimaciones iniciales, con las reales, tan pronto como se pueda.
- ◆ Hay que reaccionar rápido ante los riesgos...

# Conclusiones (cont...)

- ◆ Los modelos de estimación de esfuerzo son útiles, pero deben ser usados con cuidado. Modelos no calibrados pueden generar errores de hasta un 500% o más.
- ◆ No hay modelo que pueda captar las características del producto, del equipo de desarrollo, etc.
- ◆ Una buena intuición soportada por experiencia documentada (información histórica) generalmente es más precisa que cualquier modelo.