

# ¿Por qué fracasan los proyectos Software?

## CSTIC 2009

VII Congreso del Comité de Calidad en los Sistemas,  
Tecnologías de la información y las Comunicaciones

Madrid, 29 de septiembre 2009

ETSI Industriales-UPM (C/ José Gutiérrez Abascal, 2)



**Cristina Rivas Vila**  
Directora de Calidad y Procesos  
Application Software Group  
ALCATEL-LUCENT

**Juan Antonio López**  
Director de Calidad  
GTD SISTEMAS DE INFORMACIÓN

**Jose Barato**  
Director Asociado  
Responsable IT Governance  
ATOS CONSULTING

# Problemática: *"Algo pasa con el Software..."*

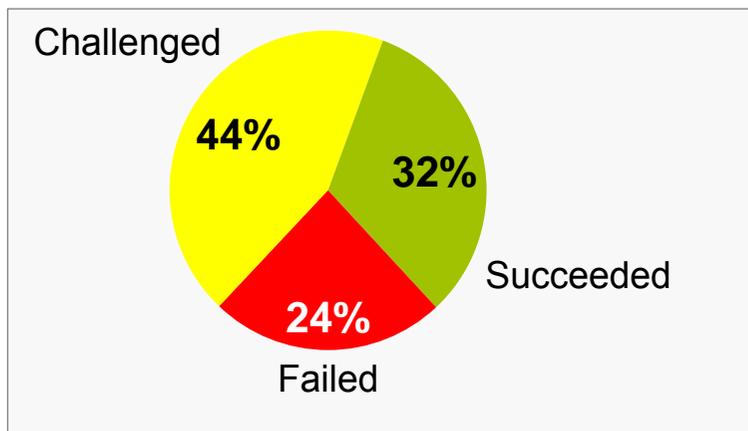
**¿Por qué fracasan los proyectos Software?**



# Algo pasa con el Software...

Uno de cada 7 proyectos se cancelan anticipadamente. La causa más común suele ser la imposibilidad de cerrar los requisitos. Para los que no se cancelan, el riesgo es alto

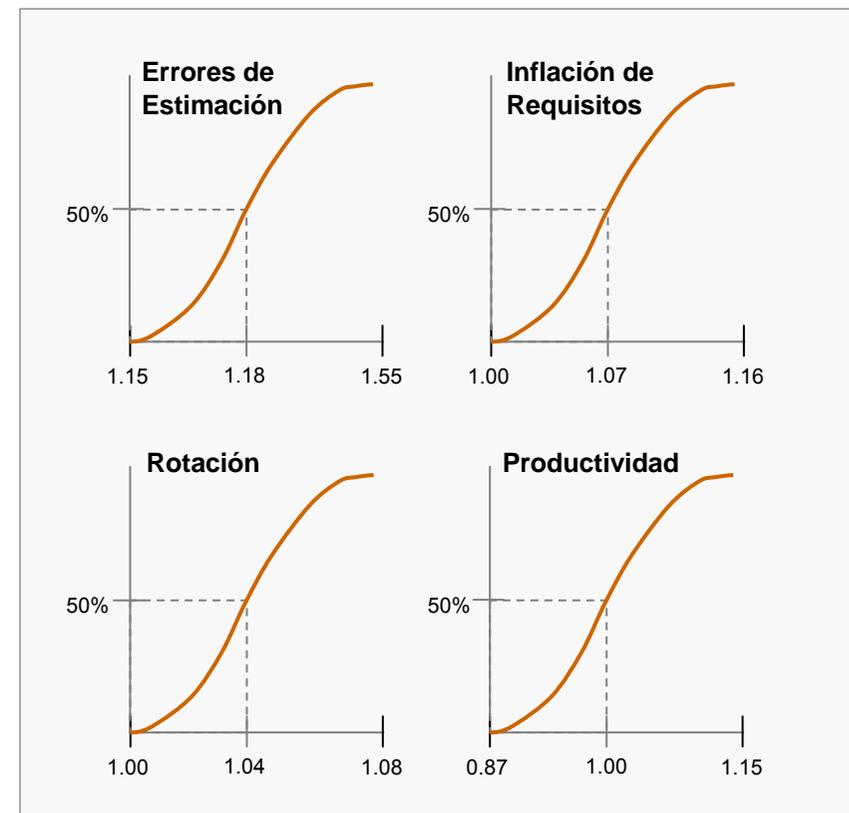
(Tom DeMarco)



**The Chaos Report:** "En media un proyecto SW se retrasará de 6 a 12 meses, y costará un 50–100% más"

- |                      |                |
|----------------------|----------------|
| 1. Requisitos        | 6. Proveedores |
| 2. Adecuación        | 7. Política    |
| 3. Entorno cambiante | 8. Conflicto   |
| 4. Recursos          | 9. Innovación  |
| 5. Dirección         | 10. Escala     |

Las 10 principales fuentes de riesgo en proyectos SW



Factor multiplicativo sobre el plazo estimado

Tom DeMarco. *Waltzing with Bears: Managing Risk on Software Projects*



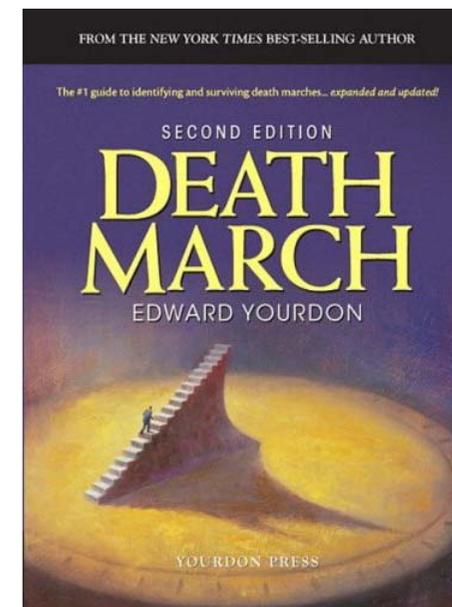
# Algo pasa con el Software...

## Proyectos en “marcha fúnebre”

Un proyecto “Death March” es aquél que excede las estimaciones en uno de sus parámetros al menos un 50%

- la mitad del **plazo**
- la mitad de las **personas**
- la mitad del **presupuesto**
- el doble de la **funcionalidad**

Tipo	Personas	Duración
▪ Pequeños	<10	3 – 6 meses
▪ Medianos	20 – 30	1 – 2 años
▪ Grandes	100 – 300	3 – 5 años
▪ Desorbitados	1000 – 2000	7 – 10 años



*Death March, the Complete Software Developer's Guide to Surviving "Mission Impossible" Projects*

*Edward Yourdon*

*Prentice Hall PTR, 2004*





## Algo pasa con el Software...

**Objetivo: Que la ingeniería del Software tan gestionable, predecible y repetible, como las demás, pero...**

- El software “no se ve”
- Los productos software son muy complejos
- Es una disciplina aún joven
- No es, ni será nunca, una ciencia exacta
- Tiene un marcado carácter “sociológico”



# ¿Por qué algunos proyectos fracasan y otros son un éxito?

La planificación básica del proyecto, la buena comunicación entre TI y Negocio, y las habilidades en gestión y liderazgo son los prerequisites básicos para conseguir mejores resultados en los proyectos de TI (Gartner)

## Gartner



From the CIO Trenches: Why Some Projects Fail and Others Succeed (Gartner, September 2007)



# Causas y Soluciones

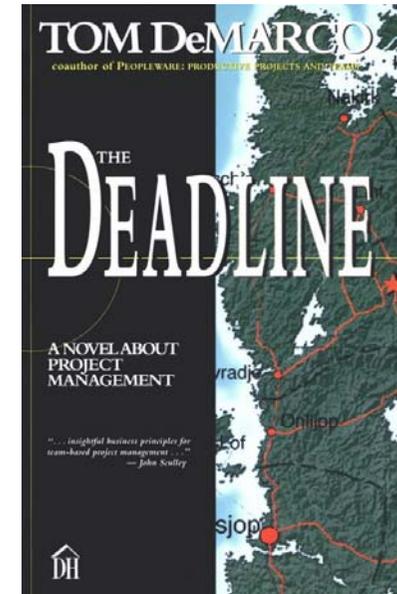
**¿Por qué fracasan los proyectos Software?**



# El lado humano del Software

## Gestionar personas es lo más importante (soft skills)

- La mayor parte de los proyectos que fracasan lo hacen por razones sociales, no por razones técnicas
- Los 4 puntos esenciales de la buena gestión
  - Contratar a las personas adecuadas
  - Asignarlas a los trabajos adecuados
  - Mantenerlas motivadas
  - Ayudar a sus equipos a integrarse y permanecer integrados
- Las partes del cuerpo esenciales de un Director de Proyectos son: tripas, corazón, alma y olfato:
  - Confía en tus **tripas** (tu instinto),
  - Lidera con el **corazón**
  - Construye **alma** en la organización
  - Desarrolla **olfato** para los problemas



*The Deadline: A Novel About Project Management*

*Tom DeMarco*

*Dorset House Publishing, 1997*





# El lado humano del Software

¿Se puede industrializar el desarrollo de SW?

Sí, pero... programar es un trabajo intelectual

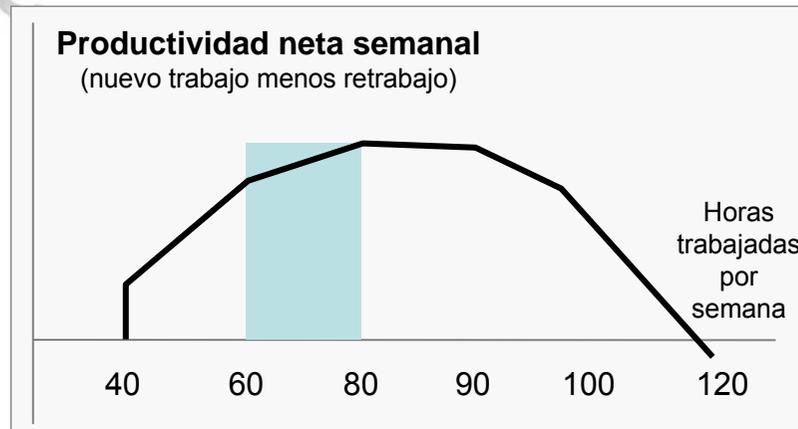
“Producir y vender hamburguesas”	Prestar servicios Software
<ul style="list-style-type: none"><li>» Reducción de la tasa de error</li><li>» Dirección autoritaria (presionar, más horas)</li><li>» Recursos intercambiables</li><li>» Ritmo de producción constante</li><li>» Proceso estándar</li></ul>	<ul style="list-style-type: none"><li>» Exigir una cuota de error (poder de veto)</li><li>» A los programadores les gusta su trabajo</li><li>» Los programadores no son intercambiables</li><li>» Un proyecto es algo dinámico</li><li>» Replantearse el trabajo (no sólo hacerlo)</li></ul>

*Tom DeMarco & Timothy Lister. Peopleware: Productive Projects and Teams*

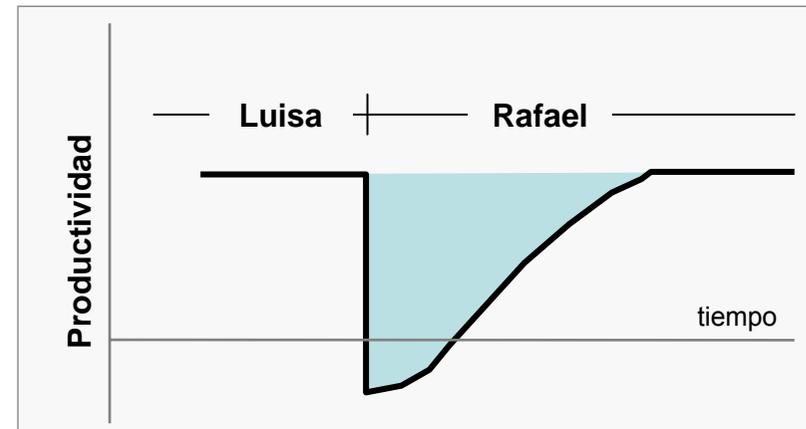


# El lado humano del Software

En los proyectos software, la Productividad depende fuertemente de las personas



Trabajar más de 80 horas semanales no mejora la productividad



Producción perdida debido a una sustitución del personal

*Tom DeMarco & Timothy Lister. Peopleware: Productive Projects and Teams*



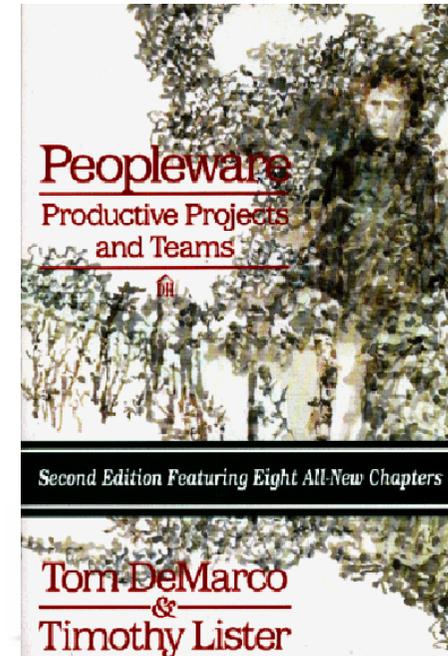


# El lado humano del Software

## Construir equipos integrados

- En los proyectos de SW no hay *team building*, sólo hay *team growing*
  - Sólo puedes facilitar y dejar que pase... ¡Si tienes suerte!
- Sí es posible asegurar cuándo no hay que esperar un equipo cohesionado como resultado de un proyecto
- Culpables de “teamicidio”
  1. El jefe que no confía en su equipo
  2. El papeleo
  3. La separación física
  4. La multitarea
  5. La reducción auto-impuesta en la calidad
  6. Los plazos imposibles (ley de Parkinson)
  7. El control sobre las camarillas
  8. Las horas extra (salvo para algunos)

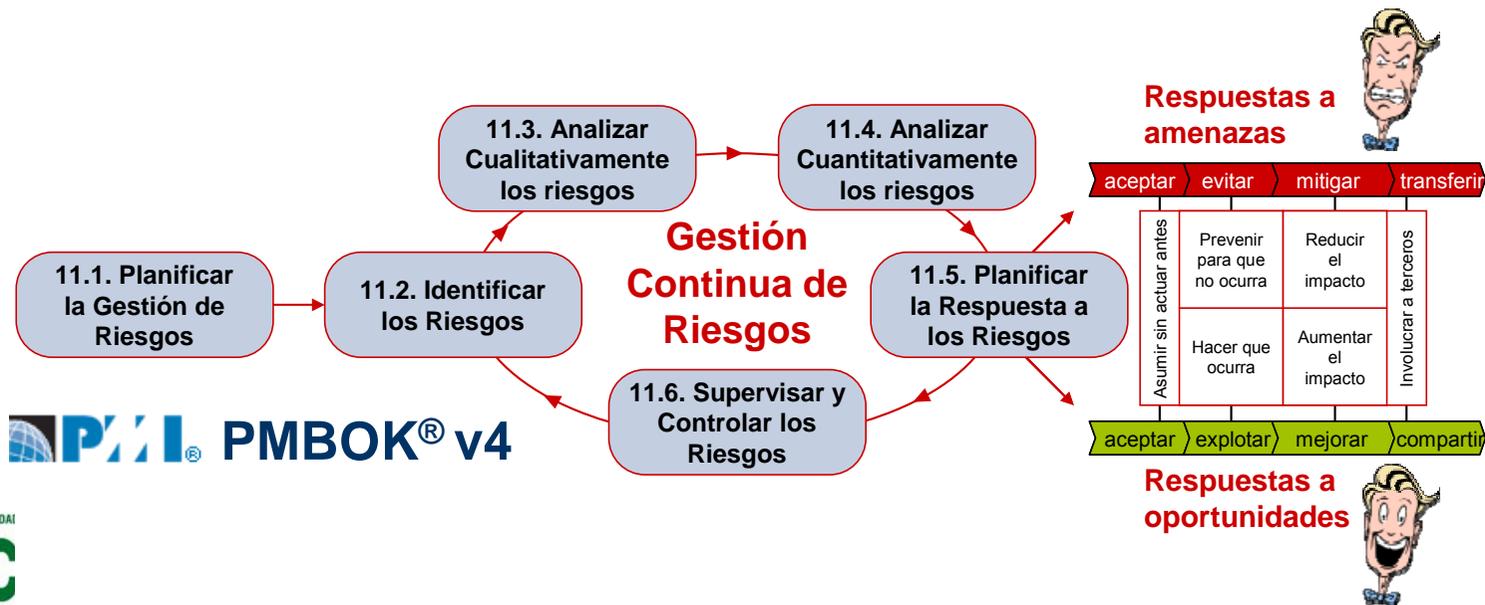
**Teamicide**



# Gestión de Riesgos

*No somos tan malos estimando: Sobrecoste y retraso se deben generalmente a tareas imprevistas*

- Un riesgo es un problema que aún no ha ocurrido. Un problema es un riesgo que se ha materializado.
- Gestión de Riesgos no es lo mismo que “preocuparse por el proyecto”.
- Gestión de Riesgos es Gestión de Proyectos para adultos
- Lo contrario a Gestión de Riesgos se llama Gestión de Crisis (intentar descubrir qué hacer con los problemas después de que ocurren).

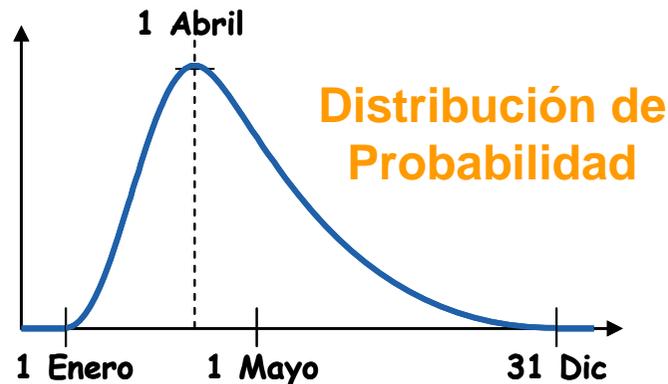


**PMI** PMBOK® v4

# Gestión de Riesgos

## Cuantificar la incertidumbre con diagramas de probabilidad

- La gráfica permite representar visualmente:
  - Terminar antes de enero es imposible.
  - La fecha más probable para entregar un producto aceptable es el 1 de abril, pero incluso esta fecha no es muy creíble (30%).
  - Hay que esperar a mayo si se quiere publicar una fecha límite con el 50% de confianza.
  - Si se quiere probabilidad de retraso virtualmente nula, hay que publicar como fecha de fin de proyecto el 31 de diciembre.

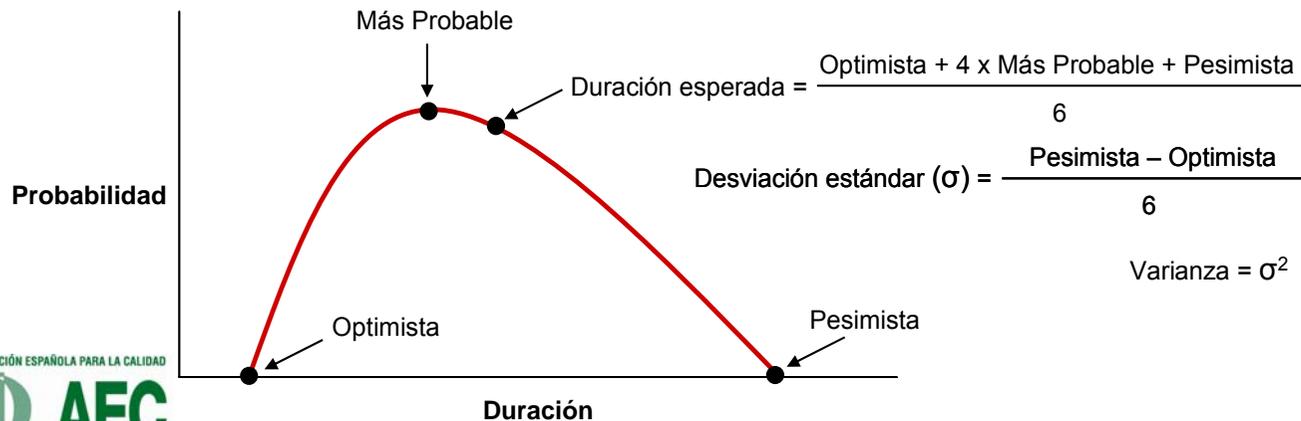


# Gestión de Riesgos

Quantificar la incertidumbre con diagramas de probabilidad

Una estimación sin  $\pm$  no es una estimación

- Media ponderada = (pesimista + 4\*más probable + optimista) / 6
- Desviación estándar ( $\sigma$ ) = (pesimista – optimista) / 6
- Hay un 68% de probabilidad de terminar en  $\pm 1 \sigma$
- Hay un 95% de probabilidad de terminar en  $\pm 2 \sigma$
- Hay un 99% de probabilidad de terminar en  $\pm 3 \sigma$
- A mayor  $\sigma$ , mayor riesgo





# Gestión de Requisitos

- Hay proyectos que comprometen trabajo con clientes (internos o externos) que no están suficientemente definidos.
- Otros esperan hasta tener un contrato o una orden de compra para discutir detalles que son cruciales para la arquitectura y la planificación del proyecto.
- En algunos casos, el proceso de negociación es largo y los cambios son numerosos durante este periodo.
- Los equipos de ventas manejan la negociación y aceptan cambios que no son suficientemente supervisados desde el equipo de producto.

## Recomendaciones

- Los equipos de preventa deben documentar todos los compromisos y mantenerlos actualizados en "tiempo real" en un lugar accesible, con control de cambios y con capacidad de trazabilidad a funcionalidad del producto.
- Debe existir un proceso de gestión de cambios durante la fase de preventa que permita que las negociaciones sean ágiles sin perder el control sobre el contenido de los compromisos.





# Gestión del Cambio

- Es un hecho que los proyectos cambian durante su ciclo de vida.
- Sin embargo, es común encontrar proyectos en los que no hay un proceso definido para gestionar el cambio o si lo hay, no es un proceso E2E sino que está limitado a una parte de la organización.
- Muchos proyectos no miden el cambio durante la vida del proyecto y no analizan sus repercusiones en fechas, costes y calidad.

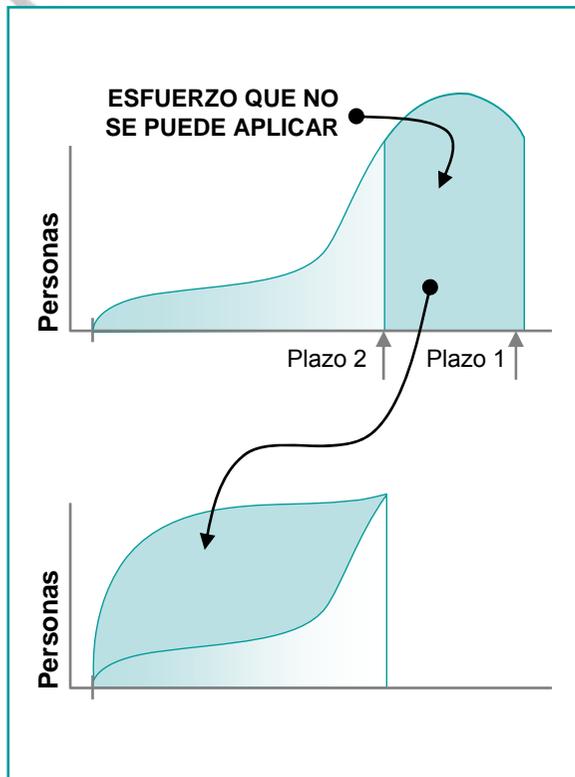
## Recomendaciones

- Proceso de cambio documentado y acordado (incluso con cliente) desde día 0.
- Gestión de riesgos asociada con cada cambio solicitado.

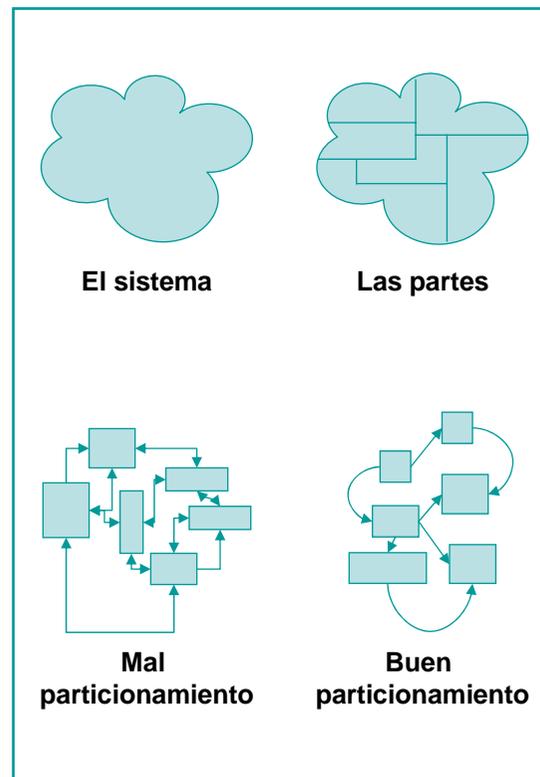


# ¿Alguien ha visto un buen diseño?

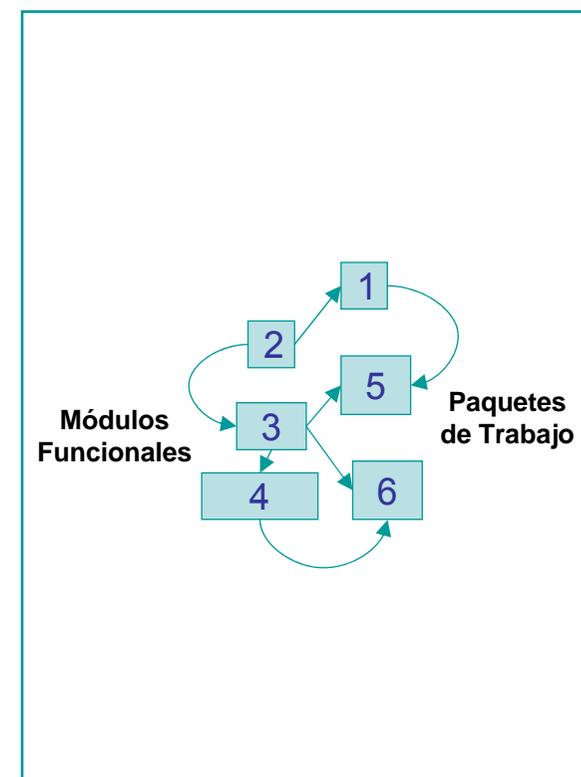
Si se cambian meses por personas, el resultado es el no-diseño  
El diseño está en el camino crítico, antes que el desarrollo  
La clave de un buen diseño es un buen particionamiento



El diseño está en el camino crítico Si se hace a la vez que el desarrollo, entonces no hay diseño



¿Alguien ha visto un buen diseño?



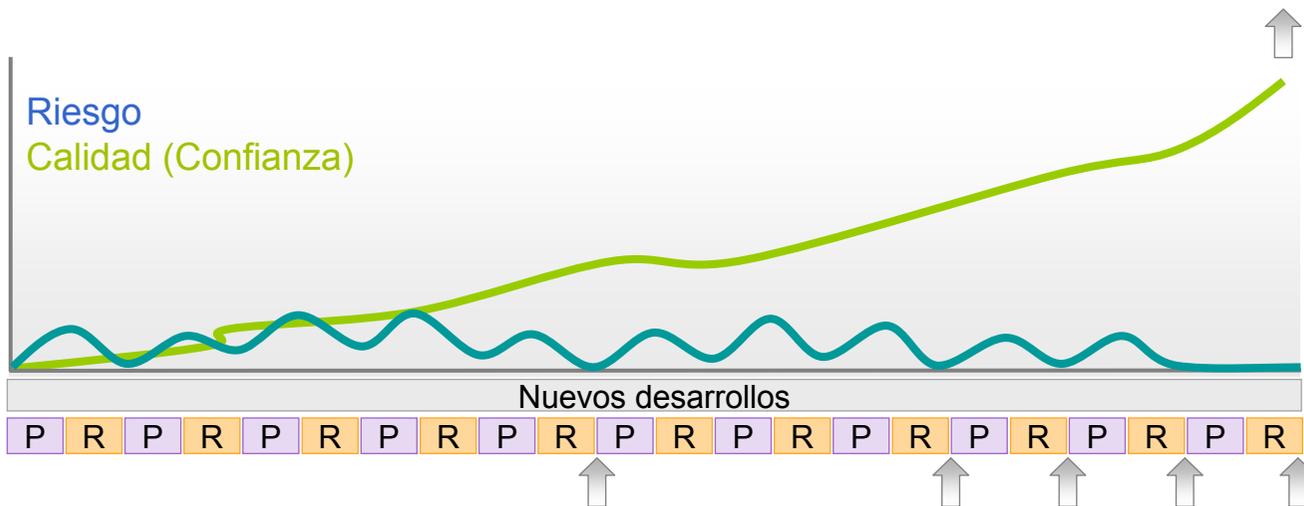
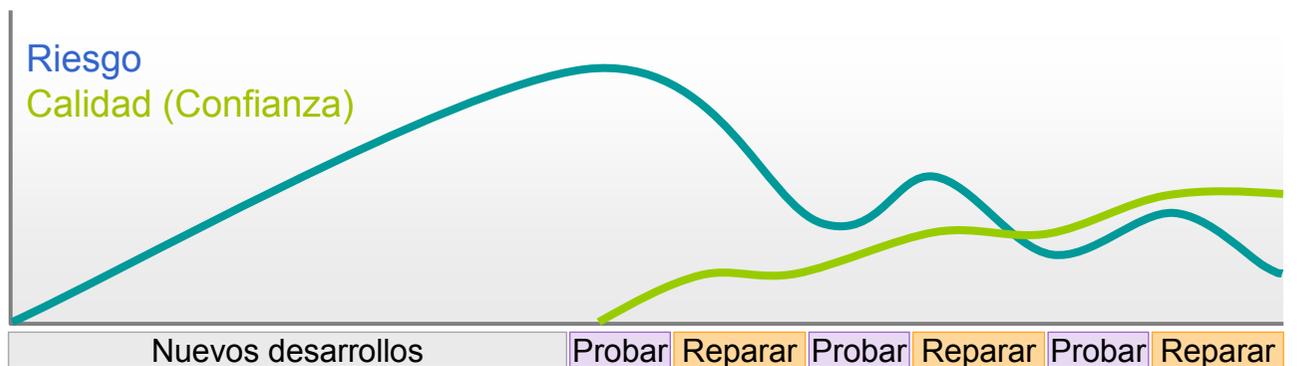
Órdenes de trabajo isomorfas con el diseño + certificaciones parciales 17



# Practicar iteraciones

El software “no se ve”. Los proyectos de más de 3.000 hh acumulan mucho riesgo

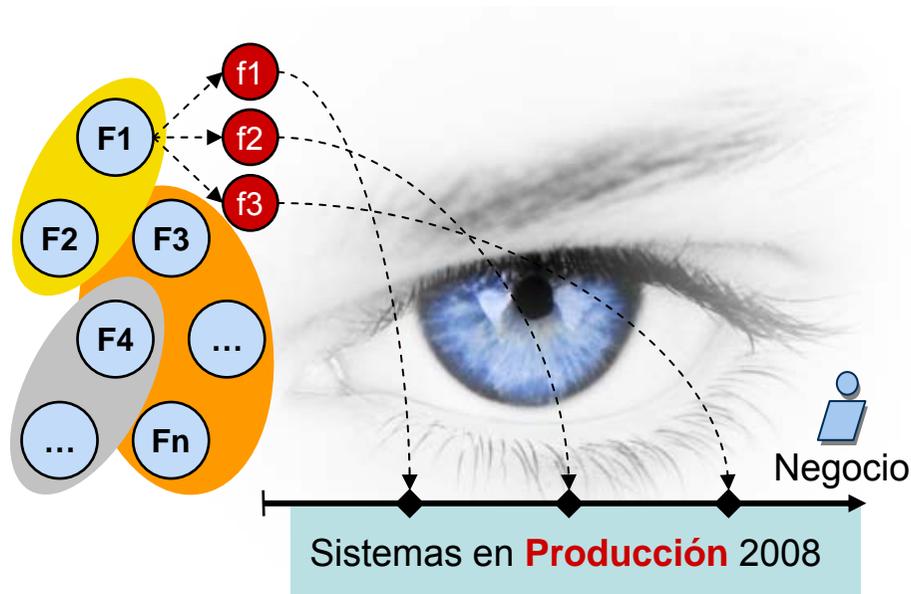
Una buena solución es practicar certificaciones parciales (iteraciones incrementales)



# Practicar iteraciones

¿“*Time to market*” como instrumento de control?

Sospecha: No se encargan evolutivos grandes a Factoría porque Negocio sólo puede “ver” el resultado tras la puesta en producción



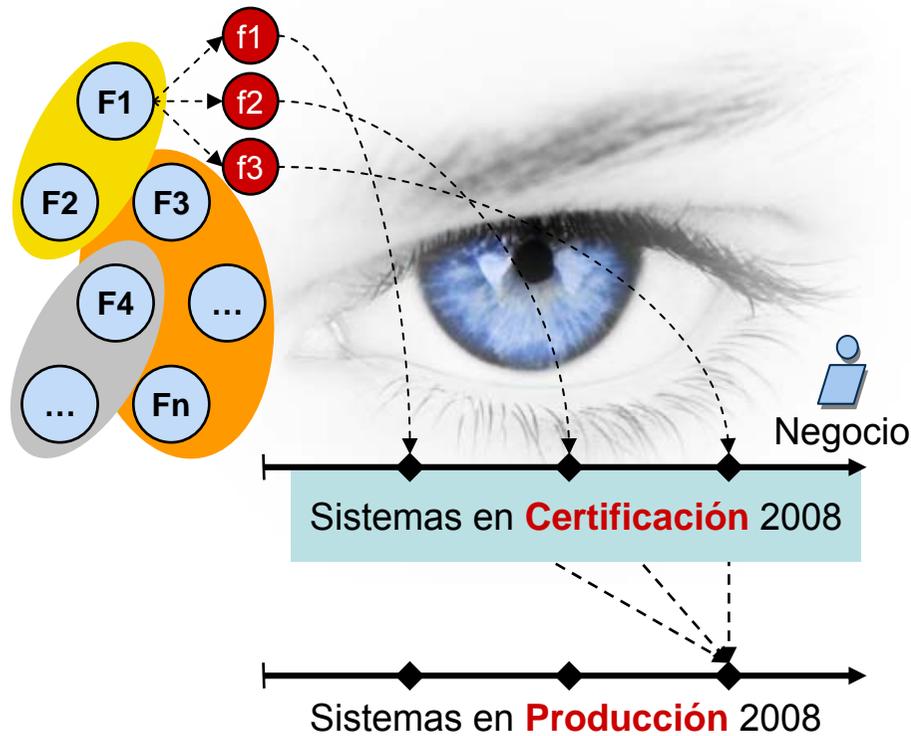
*“El trabajo se expande hasta ocupar todo el tiempo disponible.”*

*Ley de Parkinson*



# Practicar iteraciones

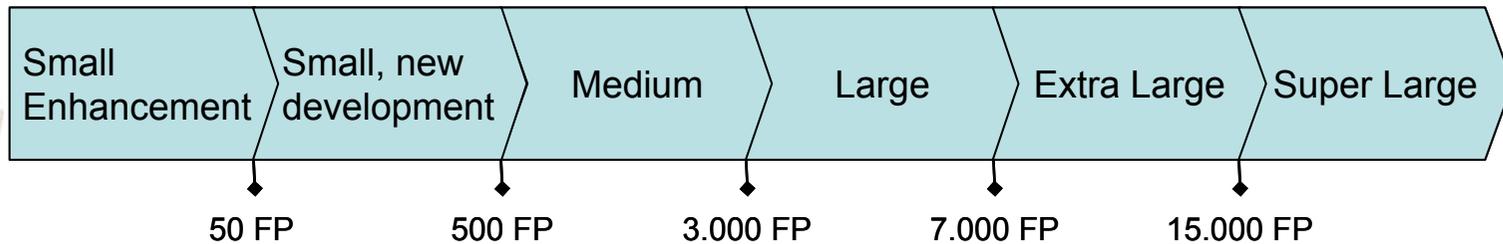
Un entorno de certificación permite validar entregas parciales sin necesidad de pasarlas a producción, y a la vez controlar el diseño



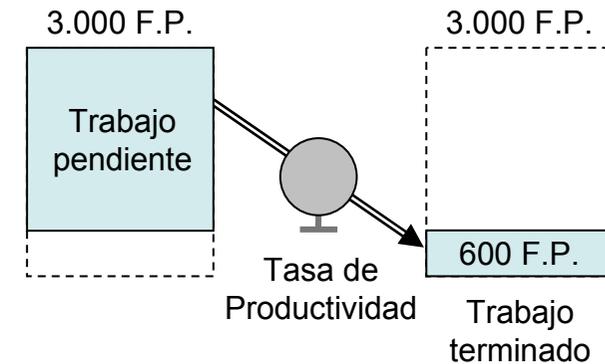
# Métricas de tamaño y productividad

- Tamaño homologable en puntos de función

Gartner



- Productividad media
  - 20-40 FP/SM <sup>(1)</sup>
  - Entre 1 y 2 FP por día y persona



<sup>(1)</sup> Unidad de esfuerzo SM (staff-month ó persona-mes) = 165 horas-hombre = 7,5 horas x 22 días



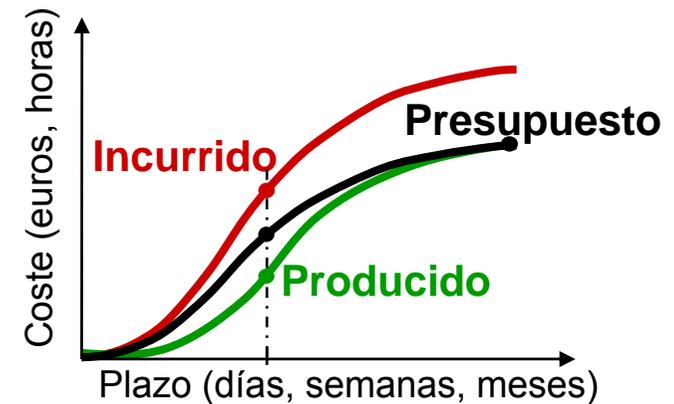


# Desviación en Plazo y Coste con EVM

## Gestión del Valor Ganado

**EVM** es un método utilizado para medir el progreso de ejecución de un proyecto de forma objetiva. Combina tres aspectos de capital importancia en la ejecución de un proyecto: técnico (cumplimiento del trabajo planificado), costes (si se gasta más o menos de lo planificado) y plazo (si el proyecto se adelanta o se retrasa).

1. La planificación detallada del proyecto indica lo que se va a hacer y en qué fechas, así como cuánto se ha pensado que costará (tanto en esfuerzo de personal como en materiales). Esto da lugar a una serie de datos que se conoce como **Valor Planificado**, que no es otra cosa que la periodificación del presupuesto.
2. Por otra parte, se determina en cada momento el grado de terminación de las actividades planificadas al comienzo del proyecto. Esto da lugar a otra serie de datos que se conoce como **Valor Ganado**, que indica el coste producido hasta la fecha.
3. Y finalmente, se conoce en cada momento el valor de lo que se ha gastado, o **incurrido** hasta la fecha.



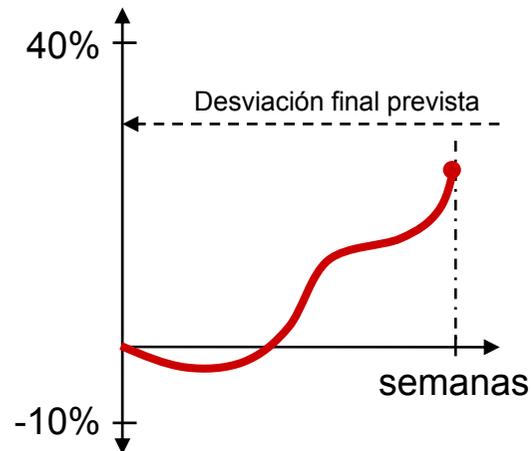


# Desviación en Plazo y Coste con EVM

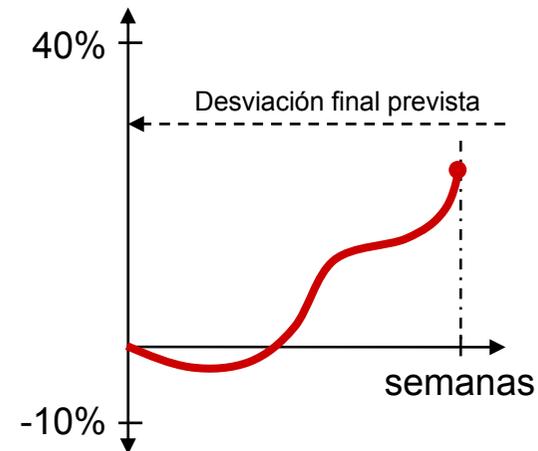
## Gestión del Valor Ganado

La técnica de Gestión del Valor Ganado es útil no sólo para representar el rendimiento actual de un proyecto, sino también para estimar las futuras desviaciones en coste y plazo

### Desviación en Coste



### Desviación en Plazo





## Usando EVM

Un proyecto tiene un presupuesto de 100 M€, duración 5 años (60 meses)  
Al finalizar el 4º año, se ha gastado 90 M€, se había presupuestado 80 M€  
El trabajo se ha terminado al 75% (se ha “ganado” 75 M€)

- $EV = 75 \text{ M€}; AC = 90 \text{ M€}; PV = 80 \text{ M€}$
- $CPI = 75 / 90 = 0,83$
- $CV = 75 - 90 = -15 \text{ M€}$
- $ETC = (100 - 75) / 0,83 = 30 \text{ M€}$
- $EAC = 90 + 30 = 120 \text{ M€}$
- $PCO = 120 - 100 = 20 \text{ M€} = 20\%$
- $SPI = 75 / 80 = 0,94$
- $SV = 75 - 80 = -5 \text{ M€}$
- $TV = (75 - 80) / (100/60) = -3 \text{ meses}$
- $TEAC = 60 / 0,94 = 64 \text{ meses}$
- $TVAC = 60 - 64 = -4 \text{ meses}$

**El proyecto tiene un sobrecoste actual de 15 M€** ----->

**El sobrecoste previsible al final del proyecto será de 20 M€** ----->

**El proyecto tiene un retraso actual de 3 meses**

**El retraso previsible al final del proyecto será de 4 meses**

**Sarbanes-Oxley  
material  
financial  
issues**

# Conclusiones

**¿Por qué fracasan los proyectos Software?**





# Principales causas de éxito de los proyectos Software

1. El equipo cohesionado: El todo es más que la suma de las partes
2. Liderazgo
3. Alineación de perfiles técnicos y funcionales (mismo idioma)
4. La orientación al “valor para el negocio”
5. Gestión de riesgos
6. Gestión continua de requisitos y cambios: Diseño, iteraciones incrementales
7. Dominio de las técnicas y prácticas (*hard skills*)
8. Gobierno, procesos corporativos
9. Financiación
10. Patrocinio





# Principales causas de fracaso de los proyectos Software

1. Rendimiento deficiente del equipo (programadores desmotivados)
2. Política (plazos imposibles, poca financiación, inflación de expectativas, intereses contrapuestos, o contrarios a los objetivos del proyecto)
3. Gestión deficiente de requisitos
4. Gestión deficiente del riesgo
5. Gestión deficiente del cambio
6. Gestión deficiente de la calidad: Mentalidad “calidad si hay tiempo”
7. Ausencia de iteraciones (el SW no se ve hasta el final del proyecto)
8. No-diseño
9. Competencia deficiente del Director del Proyecto (liderazgo, gestión)
10. Requisitos no funcionales: Innovación, escala



# Resumen



¿Por qué fracasan los proyectos SW?

