

Causas y consecuencias  
de la implantación de  
metodologías de desarrollo y  
medidas de la calidad del  
Software en organizaciones.  
Un caso práctico

Juan Fco. Hdez. Ballesteros

Jesús M<sup>a</sup> Minguet Melián

## La situación de partida

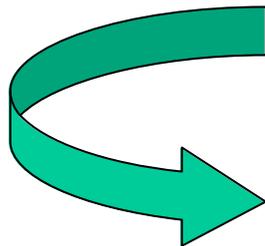
# La organización

- ✍ Organismo público.
- ✍ Número de empleados: 3000.
- ✍ Presupuesto anual: 600.000.000 €
- ✍ Ámbito de actuación: provincial.
- ✍ Número de ordenadores personales: 1500.
- ✍ Número de aplicaciones existentes en explotación: 150.
- ✍ Número de aplicaciones en desarrollo: 5.
- ✍ Número de servidores medios (valor entre 30.000 € y 120.000 €): 35.
- ✍ Número de servidores grandes (valor >120.000 €): 3.
- ✍ Personal informático: 15 técnicos de distinto nivel y formación.
- ✍ Presupuesto anual en informática: 5.000.000 €.

# La situación de partida.

## Las causas

- Inadecuada captura de necesidades funcionales.
- Personalización de la informática. Existencia de “gurús”
- Mala integración de la Unidad de desarrollo y la Unidad de sistemas. Falta de coordinación en informática.
- Falta de control del producto entregado.
- Inexistencia de mecanismos de control en el desarrollo.
- Equipo de desarrollo poco motivado.

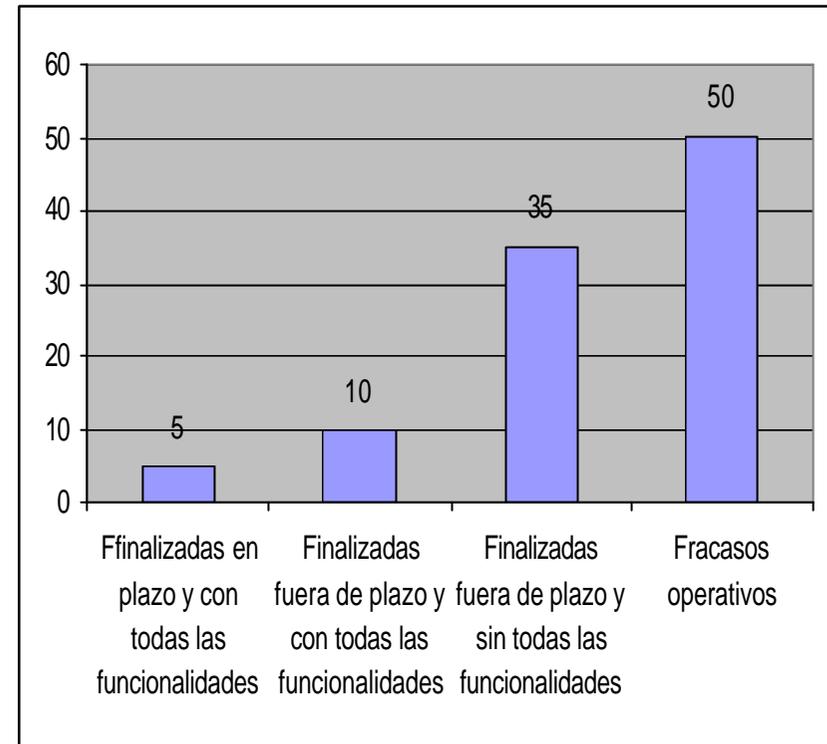


FALTA DE UNA METODOLOGÍA DEFINIDA PARA LAS  
TECNOLOGÍAS DE LA INFORMACIÓN Y LAS  
COMUNICACIONES

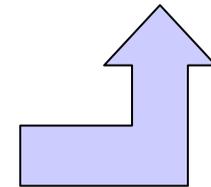
# La situación de partida.

## Las consecuencias

- Aplicaciones entregadas fuera de plazo o no finalizadas.
- Aplicaciones entregadas sin las funcionalidades mínimas.
- El equipo de desarrollo toma decisiones que no le corresponden.
- Programas informáticos con fallos en ejecución. Poco fiables.
- Costes económicos por encima de las estimaciones iniciales.
- Funcional desmotivado y reticente a nuevos proyectos informáticos.
- Usuarios poco participativos pero muy exigentes.



Aplicaciones informáticas sin calidad



## Las primeras acciones correctivas

- ◆ Se consideró necesaria la realización de un Plan de Sistemas.

# Las primeras acciones correctivas

## ◆ Implantación de metodología para desarrollo y explotación de nuevos proyectos informáticos

- Comenzar a implantar METRICA V 3.0 pero adaptando la metodología a la organización. No al revés.
- Obligatoriedad de la realización de pruebas funcionales y técnicas para aceptar las aplicaciones.
- Preparación de entornos de pruebas, desarrollo, preexplotación con procedimientos muy estrictos para modificar una aplicación en explotación.
- Cada proyecto informático debía tener su responsable funcional definido por el Servicio Administrativo o Técnico al que afectaba la aplicación “coordinador funcional informático”

# Las primeras acciones correctivas

## ◆ Implantación de metodología para desarrollo de nuevos proyectos informáticos

- Realización de un inventario de aplicaciones.
- Emitir una política de seguridad en cuanto a accesos a las aplicaciones y el procedimiento de permisos.
- Primer control de calidad tosco, pero efectivo, basado en productos finales desarrollados. Cuantificación y cualificación de las incidencias en los mismos.
- Procedimentar claramente la petición de nuevas aplicaciones exigiendo a los Servicios solicitantes una clara definición de funcionalidades, tiempos previstos de puesta en explotación y designación de un coordinador informático funcional

# Las primeras consecuencias

- ◆ Acción 1: Implantación de pruebas estandarizadas para la aceptación de aplicaciones.
  - ? No se aceptaron dos de las aplicaciones sometidas a pruebas (25%)
- ◆ Acción 2: Se exige la creación de entornos separados de pruebas, pre-explotación y explotación
  - ? Aumento del gasto en equipos informáticos.
  - ? Mejora los niveles de seguridad, pre-explotación es válido para el plan de contingencia.
  - ? Mejora sustancial en el número de incidencias (fallos operativos, funcionalidades mal definidas) de los programas en explotación. La calidad del producto aumenta desde el punto de vista del aumento en el atributo fiabilidad.
- ◆ Acción 3: Se ha de nombrar un responsable funcional por parte de los Servicios o Unidades.
  - ? Aceptación en la organización de esta iniciativa. Rechazo de Servicios más reticentes.
  - ? Se detecta un problema no identificado anteriormente. La falta de compromiso y autoexigencia de los funcionales implica que algunas aplicaciones informáticas se retrasen considerablemente o no respondan a las necesidades reales de la organización.

# Las primeras consecuencias

## ◆ Acción 4: Procedimiento de petición de aplicaciones según un protocolo definido.

- ? Avalancha de peticiones de nuevas aplicaciones informáticas. Solicitudes mal planteadas, sin definiciones medianamente claras de las necesidades a cubrir y sin determinar el coordinador informático por parte del Servicio.

## ◆ Acción 5: Implantación progresiva y limitada de METRICA

- ? En la Organización.

- ✍ Mayor control de las aplicaciones en desarrollo. Detección de desviaciones

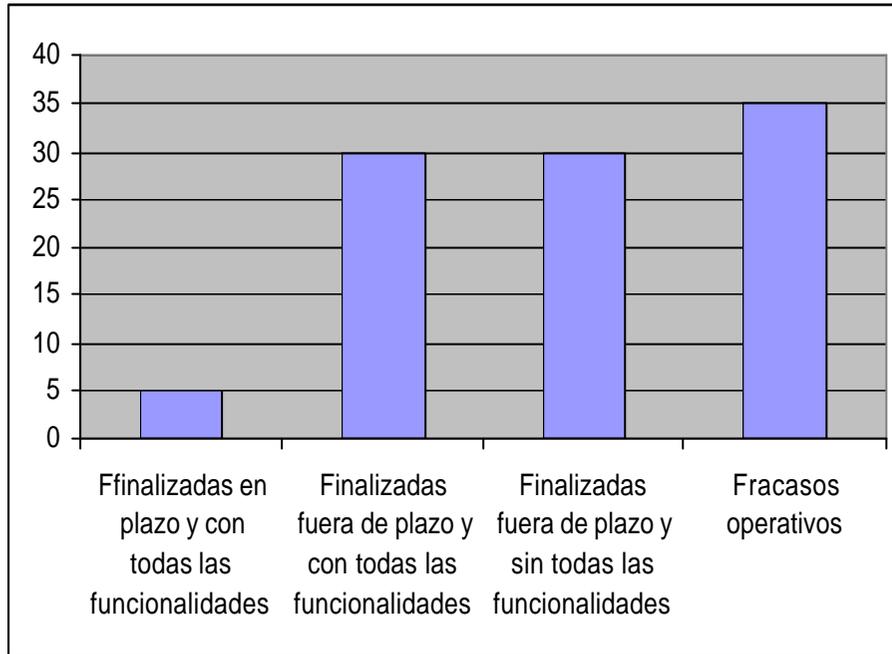
- ✍ Mejor preparación del personal de Informática.

- ? En empresas subcontratadas

- ? Mayor coste para la empresas y por tanto para la Organización.

- ? La empresas se hacen más autoexigentes.

# Las primeras consecuencias



Mayor control pero no total.

Errores importantes en la estimación de esfuerzos

✍ Disminución de los fracasos operativos (50% a 35%)

✍ Aumento del número de aplicaciones finalizadas con todas sus funcionalidades pero fuera de plazo (10% a 35%)

- Se exigía que las aplicaciones dieran respuesta a necesidades claramente establecidas por los Servicios. Se sabía qué se quería.

- No había control total en el plazo de entrega. Las empresas y el Servicio de Informática no sabían cuantificar y determinar el tiempo de implantación de un proyecto informático.

✍ Otras estadística no tan positivas debido al tiempo transcurrido desde la adopción de medidas (1 año aprox.) (5% fracasos operativos, 30% finalizadas sin todas las funcionalidades y fuera de plazo)

# Mayor conocimiento del Software. Mayor control

➤ ¿Qué ha fallado para no controlar el desarrollo software y obtener productos en forma y tiempo adecuado?

☑ Falta de preparación de los técnicos de la Organización. No se puede implantar la metodología en toda su extensión.

☑ Falta de preparación de los técnicos de las empresas subcontratadas. Claro desconocimiento de metodologías estándares.

☑ Falta de autoexigencia en los funcionales.

☑ Falta de control en el proceso software por falta de obtención de medidas fiables.

Se opta por establecer una política de medidas. Se adopta la descomposición en árbol o jerárquica y apoyándonos en ISO 9126.

# Medir el software

- ✿ Establecer una política de medidas. Situación.
- ✿ Formación del personal. Generalmente nivel muy bajo de conceptos y menor aún en el campo de aplicaciones prácticas.
- ✿ Definición de una estrategia de medidas comenzando por medidas simples del producto informático.

- Fiabilidad

$$\frac{\text{número de defectos descubiertos en el programa}}{\text{tamaño del programa}}$$

- Asignar recursos técnicos y humanos a la medida de aplicaciones.
  - Medir es algo más que la toma de datos incontrolado. Es establecer los procedimientos claros para la toma regular de datos y su explotación con el objetivo de ayudar a la toma de decisiones.
- ✿ Situación actual y resultado en la toma de medias del producto.
    - Datos obtenidos poco significativos. Se han de recopilar datos con objeto de establecer tendencias basadas en datos históricos.
    - Medidas dependientes del observador. Incoherencia conceptual.
    - Mayor conocimiento de la aplicación. Conocimiento no estricto pero válido.

# Conclusiones

- ✍ La metodología para el desarrollo de aplicaciones es una necesidad incuestionable.
- ✍ La medida de las aplicaciones también.
- ✍ El control del desarrollo del software se alcanza con el uso de metodologías apoyadas en la realización de medidas.