



¿Porqué fallan los proyectos de software?

Omar Salvador Gómez Gómez

Seminario de Software



Agenda

- Introducción
- Propiedades del software
- Los tres principios de la administración del software
- ¿Porqué fallan los proyectos de software?
- Conclusiones



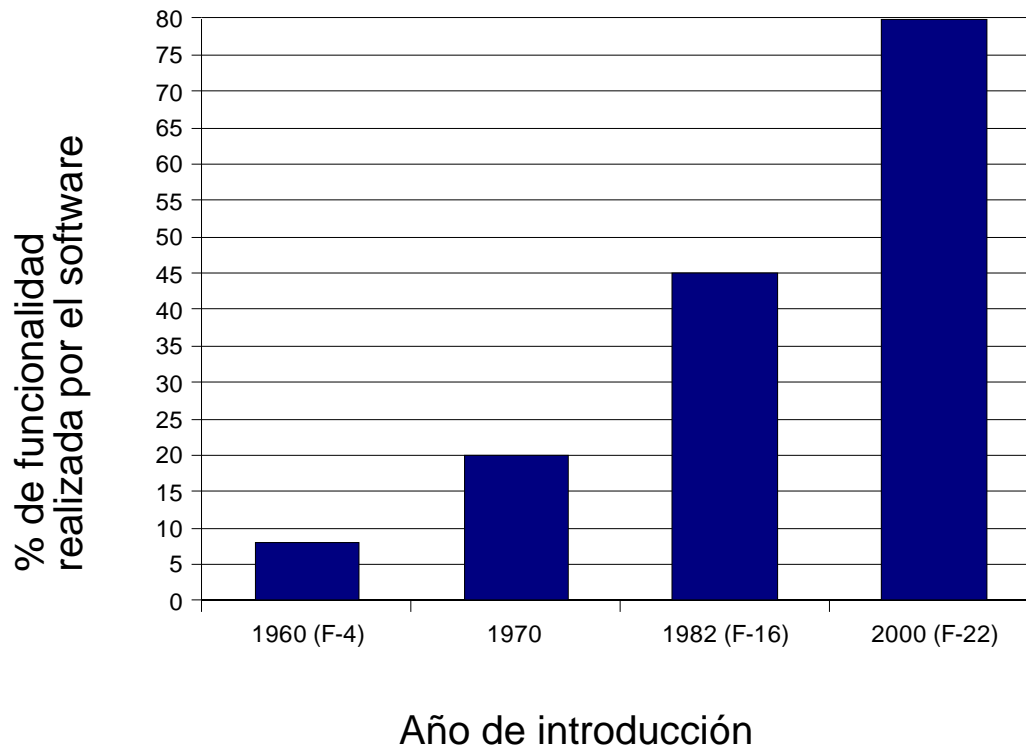
Introducción

- Cada vez más el software es parte de nuestra vida diaria, lo podemos encontrar en:
 - Computadoras
 - Teléfonos celulares
 - Autos
 - Aparatos electrónicos



Introducción

- Crecimiento del software en la fuerza aérea





Propiedades del software

- Complejidad
 - Gran cantidad de estados
- Conformidad
 - Es diseñado por diferentes personas
 - Ajustarse a diferentes software
- Cambiante
 - Sujeta a presiones de cambio
 - Nuevos usos surgen del utilizarlo
- Invisible
 - Diferentes interpretaciones



Los tres principios de la administración del software

- Reconocer que estamos en el negocio del software
- La calidad debe ser la principal prioridad
- La calidad del software es desarrollada por personas disciplinadas y motivadas



Reconocer que estamos en el negocio del software

- Actualmente en la mayoría de los negocios el software juega un papel importante
- IBM no comprendió este primer principio
- Cuando se introdujeron las PCs, IBM era la tercera compañía mas rentable del mundo, Microsoft no estaba en la lista de las compañías mas rentables
- Diez años mas tarde, IBM tuvo la pérdida de operaciones mas grande de toda la historia, Microsoft ha crecido mas del doble que IBM



La calidad debe ser la principal prioridad

- Cuando la calidad no es manejable los proyectos de software no son manejables
- Ashton Tate no comprendió este principio
- Inicio operaciones en el año de 1980 con el producto Dbase
- En 1987 era una de las tres compañías mas grandes
- Anunciaron su nuevo producto Dbase IV para mediados de 1988
- En 1991 aun continuaban reparando defectos en la fase de pruebas
- No se preocuparon por la calidad en su producto



La calidad del software es desarrollada por personas disciplinadas y motivadas

- El desarrollo de software es un trabajo intelectual
- Las personas deben comprometerse con su trabajo
- Deben ser conscientes de la calidad con que producen sus productos
- Este principio se logra únicamente por personas capacitadas y motivadas



¿Porqué fallan los proyectos de software?

- Las cinco principales causas por las que fallan los proyectos de software son:
 - Los tiempos de entrega no son realistas
 - Contar con el Staff Inapropiado
 - Cambio en los requerimientos
 - Trabajo de baja calidad
 - Creer en la magia



Tiempos de entrega no realistas

- El principal error es cambiar la estrategia de desarrollo de manera agresiva
- Lo que resulta en una mala especificación en requerimientos
- Se produce superficialmente un diseño
- Los miembros del equipo directamente empiezan a codificar



Tiempos de entrega no realistas

- Resultados:
 - Los costos de reparar defectos en la fase de pruebas pueden ser mas de 100 veces que el costo de repararlos en la fase de diseño
 - El proyecto no se entrega a tiempo
 - Se entrega el producto con una calidad pobre



Contar con el Staff inapropiado

- Asignar a los miembros del equipo más de dos proyectos a la vez
- Falta de motivación
- Falta de capacitación



Cambio en los requerimientos

- No existe administración de los requerimientos
- No se evalúa el impacto al haber algún cambio en los requerimientos
- Nunca deja de haber requerimientos a lo largo del proyecto



Trabajo de baja calidad

- Omitir las fases de requerimientos y diseño
- Intentar remover los defectos en la fase de pruebas



Crear en la magia

- Hacer uso de COTS, pensando en que solucionarán gran parte del desarrollo
- La esperanza no es una estrategia



Algunos proyectos que han fracasado

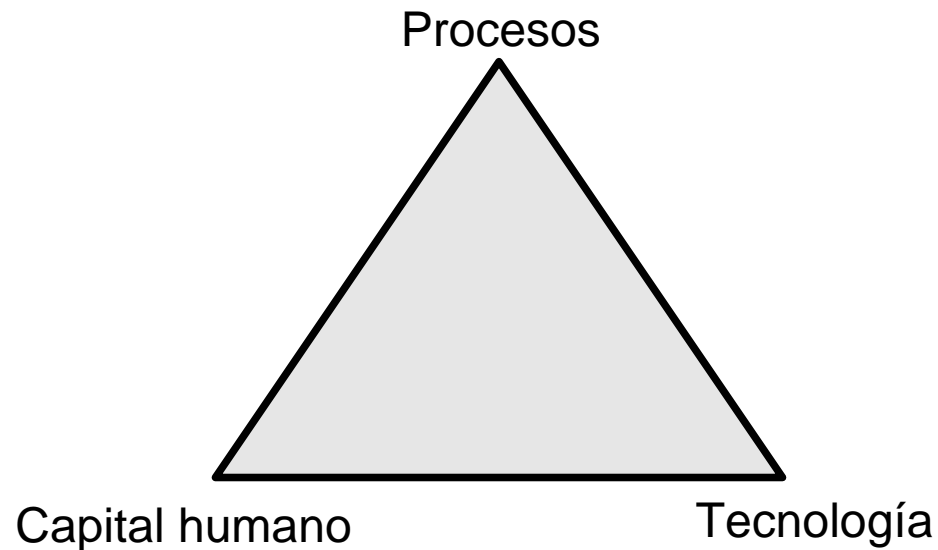
Año	Compañía	Resultado
2005	Hudson Bay Co. [Canada]	Problemas con el sistema de inventarios contribuyen a perdidas por \$33.3 millones*
2004	Ford Motor Co. [US]	Un sistema de compras es abandonado después de gastar \$400 millones* en desarrollo
2002	McDonald's Corp. [US]	Se cancela un sistema innovador de compras después de gastar \$170 millones*
1996	Arianspace [France]	Errores en la especificación y diseño del software ocasionan la explosión del cohete Arian 5 ocasionando perdidas por más de \$350 millones*
1993	London Ambulance Service [UK]	Se cancela un sistema después de gastar mas de \$26 millones*

* Dólares



Conclusiones

- El triangulo del desarrollo de software





Referencias

- Brooks, Fredrick P. 1987. No Silver Bullet: Essence and Accidents of Software Engineering. *IEEE Computer*, 10-19.
- Watts S. Humphrey. 2002. *Winning With Software: An Executive Strategy*. Edited by The SEI Series in Software Engineering: Addison Wesley.
- Leishman, Theron R. and David A. Cook. 2002. Requirements Risks Can Drown Software Projects. *CrossTalk* (Risky requirements): 4-8.
- Charette, Robert N. 2005. Why Software Fails. *IEEE Spectrum*, 42-49.