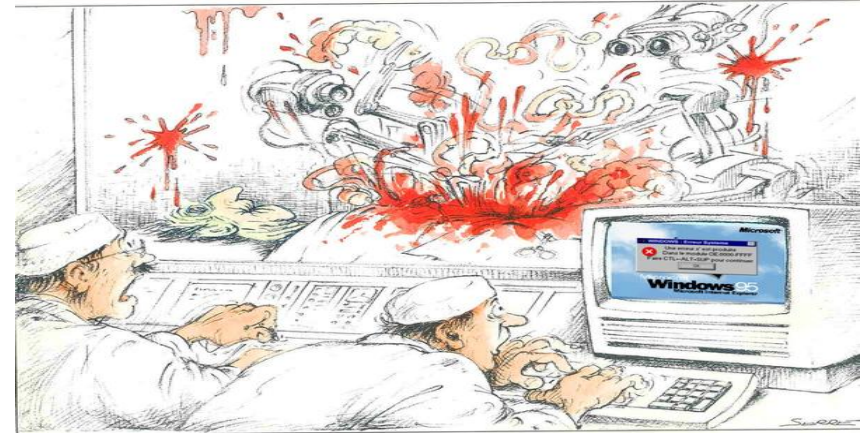


Ingeniería del Software

Desarrollo del software – Peligros
¿Qué ventajas aporta una buena metodología de desarrollo?

NOUVEAU : LE ROBOT CHIRURGIEN



Errores de Software: Comunicaciones



(1980) Arpanet

- Mutación de bits en memoria
- Código de detección de error usado en transmisión pero no en almacenamiento
- Algoritmo garbage-collection no borra mensajes duplicados
- **Caída TODA la red 4 horas**

Errores de Software: Comunicación



(1990) AT&T

- Una mejora del software de 114 switches trajo el siguiente efecto:
 - Se cae un switch, avisa a sus vecinos y se autoresetea
 - Rearranca y avisa a sus vecinos
 - Debido a un error en el software, los vecinos entran en un ciclo similar
 - El fallo se reproduce una y otra vez...
 - 5 Millones de llamadas bloqueadas. 9 horas de colapso

Errores de Software: Industria Aeroespacial



(1996) Ariane 5

- Se produjo una excepción no manejada por ADA al no poder convertir un valor de 64 bits a 16 bits
- La subrutina había sido reutilizada
- En su lanzamiento, tras alcanzar 3700m. se desvió, se rompió y estalló

Errores de Software: Defensa



(Guerra del Golfo) Misiles Patriot

- Su precisión debía ser del 95%. Se quedó en el 13%.
- Funcionando 100 horas se acumuló un error en el reloj (en pruebas sólo se había tenido 14 horas)
- Dos versiones del número 0.1 (una de 48 bits y otra de 24 bits). No son idénticas.
- No hizo blanco a un scud que mató a 28 soldados americanos

Errores de Software: Defensa



(1988) Misil Aegis Vincennes

- Un F14 se encontraba en el punto de mira. La pantalla no indicaba la altura de los objetos.
- Un objeto que subía se interpretó que bajaba.
- Se disparó el misil a un Airbus iraní (290 muertos)

Errores de Software: Aviación



(1988) Airbus A320

- Fallos de altímetro, repentinos cambios en aceleración, problemas de conducción en pista, lentitud del motor en respuesta a mandatos de pilotos, etc.
- En una exhibición murieron 3 personas y hubo 133 heridos

Errores de Software: Medicina



THERAC-25

- La interfaz gráfica permitía proporcionar dosis de radiaciones mortales a los pacientes
- 6 accidentes (tres muertes)

Errores de Software: Medicina



(1993) London Ambulance Service

- El nuevo sistema fue incapaz de asignar con rapidez ambulancias a conductores y a emergencias.
- Retrasos de 11 horas. 20 muertes por el fallo del sistema

Errores de Software

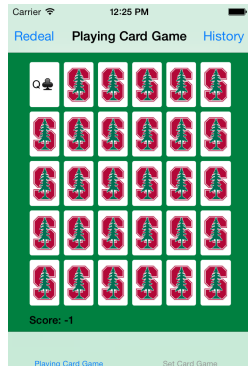
1 error cada 10 LOC

Un programa de 1.000.000 LOC tiene 100.000 errores potenciales:

- Los programadores encuentran y corrigen el 95% de los errores
- El programa final tiene 5.000 errores
- Cualquier error puede causar problemas

¿Y si utilizamos metodologías y herramientas adecuadas?

Ejemplo: Desarrollo de aplicaciones dispositivos móviles



MVC

- Modelo (Lógica de Negocio)
 - Baraja
 - Carta
 - Juego de cartas
- Vista (Interfaz gráfica)
 - Tapiz, cartas, ...
- Controlador (Interactúa con ambos)

Ejemplo: Un buen diseño del modelo

■ **Baraja** (abstracta) – conjunto de cartas

■ **Carta** (abstracta)

- **descripcion** – método que devuelve la descripción de la carta
- **emparejan** – método que, dada una lista de cartas, devuelve *true* si “emparejan” con la carta y *false* en caso contrario

■ **CartaPoker y BarajaPoker**

Ejemplo

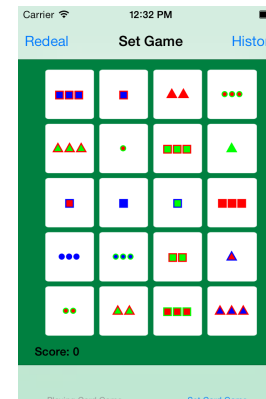
■ Suponed que hemos desarrollado un juego en el que el jugador o la jugadora debe “emparejar” cartas

- 2 o 3
- Deben coincidir en número o palo
- Baraja de **poker**

¿Y si queremos ampliar la aplicación?

Set game

- CartaSet
 - Número : 1, 2, 3
 - Forma: Rombo, Oval, Nube
 - Color: Verde, Rojo, Azul
 - Relleno: Vacío, rallado, Sólido
- Emparejan si
 - Para cada uno de los atributos
 - Las tres cartas tienen el mismo valor o las tres tienen valores distintos

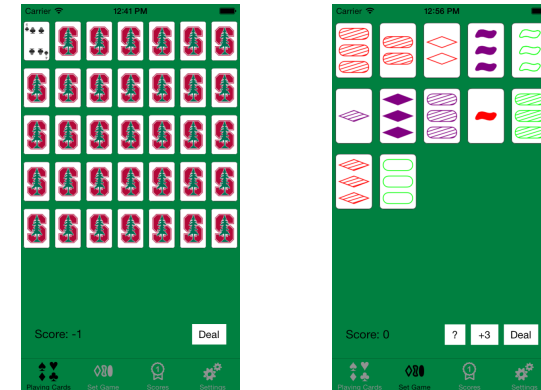


¿Cómo lo podemos conseguir?

- “Copiar” la vista
- El controlador lo podemos reutilizar
- Model
 - CartaSet
 - BarajaSet

Podemos añadir un “nuevo” juego, sin apenas desarrollar nuevo código

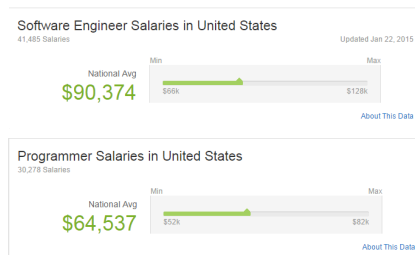
Nueva Interfaz Gráfica



Sólo habría que cambiar las vistas de las cartas

Programador vs. Ingeniero de Software

“Todo mundo” puede sentarse a programar pero esto no garantiza que pueda crear una solución compleja en tiempo, costo y calidad adecuados.”
Desarrollar software sin conocer técnicas de ingeniería representa un enfoque “artesanal” cuyo resultado son productos que no ofrecen garantías de futuro.



¿¿Cuáles son vuestras expectativas??