Testing

IIC2143

Prof. Juan Pablo Sandoval

Prueba de Conducir

¿Cuáles son los criterios que toman para otorgar la licencia?

Zig zag

Estacionamiento

Prueba Escrita

Características de la prueba de conducir

- *Las pruebas son diseñadas y preparadas.*
- *Las pruebas tienen sus objetivos pre-definidos.*
- *El test permite demostrar que el conductor satisface los requerimientos mínimos para conducir un auto.*
- *El examinador observa todo el tiempo al conductor y además se hace una prueba escrita.*

Prueba de Conducir & Software Testing

- ***Planning and reparation.*** *Ambos, tanto el examinado, como el evaluador necesitan un plan de acción y preparar de antemano el test, que no es exhaustivo, pero representativo y que permita analizar el riesgo en las decisiones.*
- ***Estático y dinámico.*** *Es útil realizar tanto tanto una evaluación dinámica (cuando conduce o cuando se ejecute el programa), como lo estático (preguntas al conductor, revisión del programa).*
- ***Evaluación.*** *Tanto el examinado como el evaluador deben realizar la evaluación de forma objetiva y reportar los resultados (observaciones) de la prueba.*
- ***Satisfacción de requerimientos.*** *Ambos analizan los resultados para ver si cumplieron los requerimientos mínimos esperados.*
- ***Detectar defectos.*** *Ambos analizan los resultados para detectar los defectos que llevaron a las fallas.*

Testing

Conceptos Básicos

Juan Pablo Sandoval

¿Qué es Testing?

Software testing es un proceso usado para evaluar la ***correctitud****,* ***completitud*** *y la* ***calidad*** *de un programa de computador. Esto puede incluir un conjunto de actividades realizadas con el fin de encontrar errores en programas de manera que puedan ser corregidos antes de lanzar el producto a los usuarios finales.*

¿Por qué el Software Testing es importante?

Software Bugs pueden ser
costosos e incluso peligrosos

Pero, ¿qué causa los defectos?

Sabemos que las personas cometemos errores, somos falibles (contrario a in-falibles). Los errores o equivocaciones pueden venir de diferente lugar. Por ejemplo:

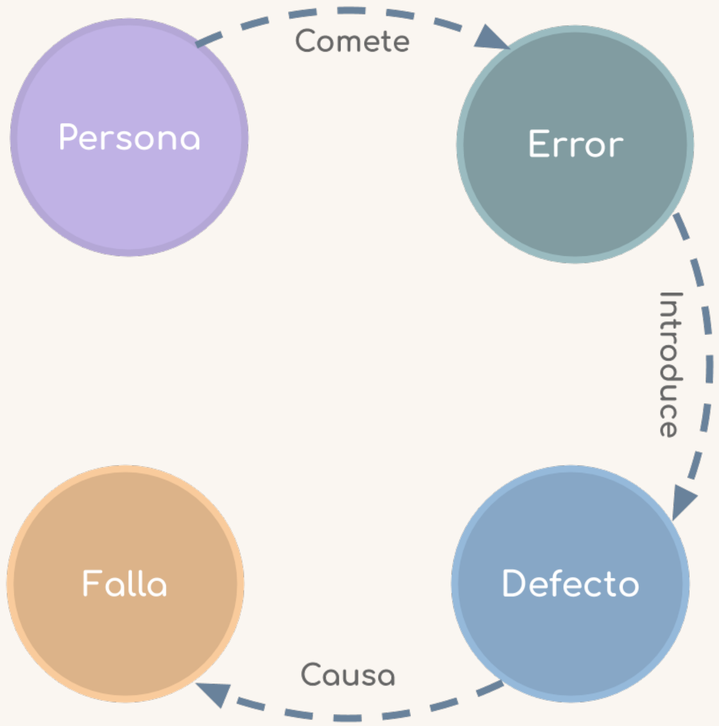
- ***programadores:*** *errores en el diseño, construcción, etc.*
- ***usuarios finales:*** *utilizan el software de forma "no esperada".*

Pero que podría malir sal

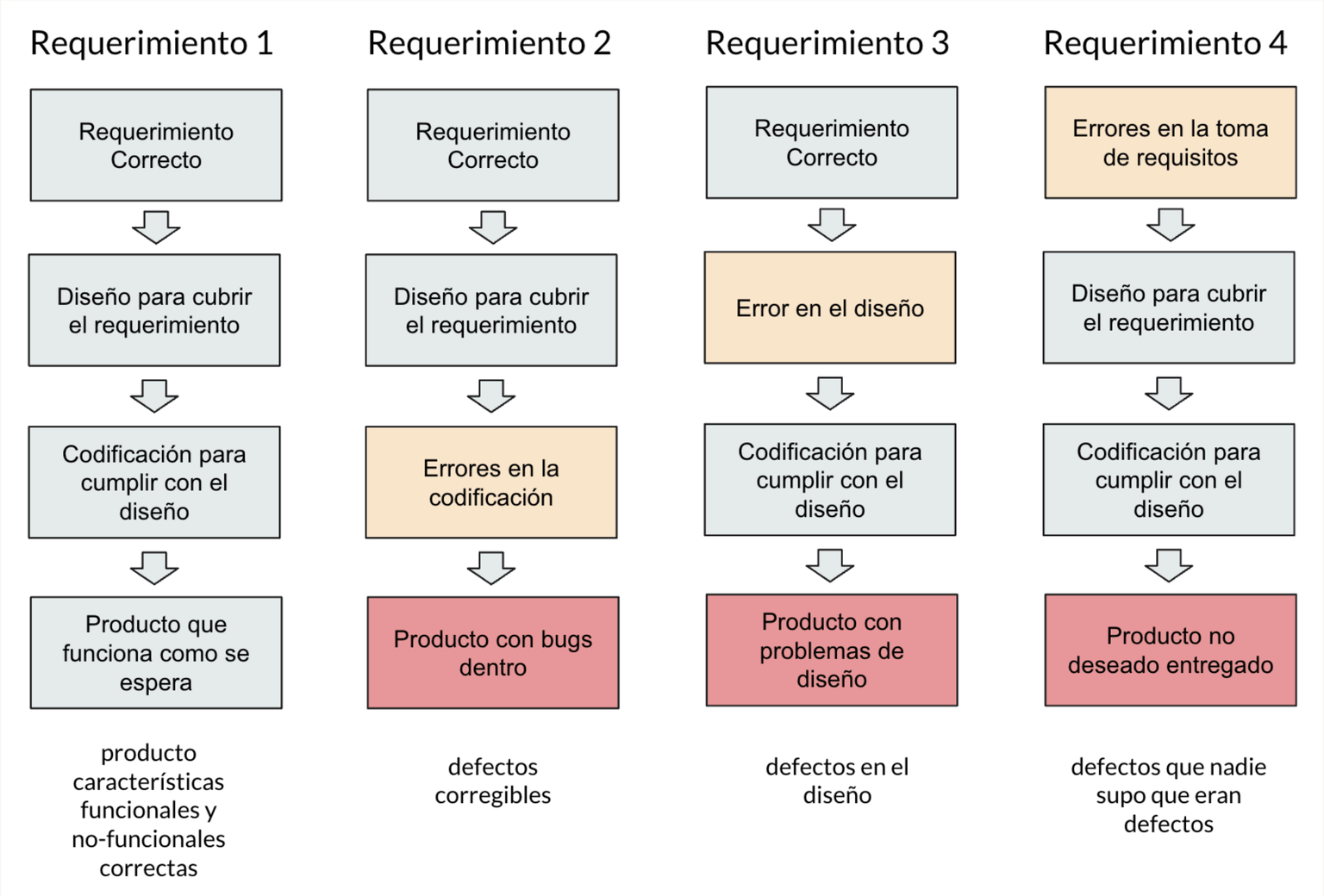
Varios factores pueden llegar a defectos o fallas:

- *Errores en la especificación, diseño, implementación*
- *Errores en el uso del sistem*
- *Condiciones del medio ambiente*
- *Daño intencional*
- *Entre otros*

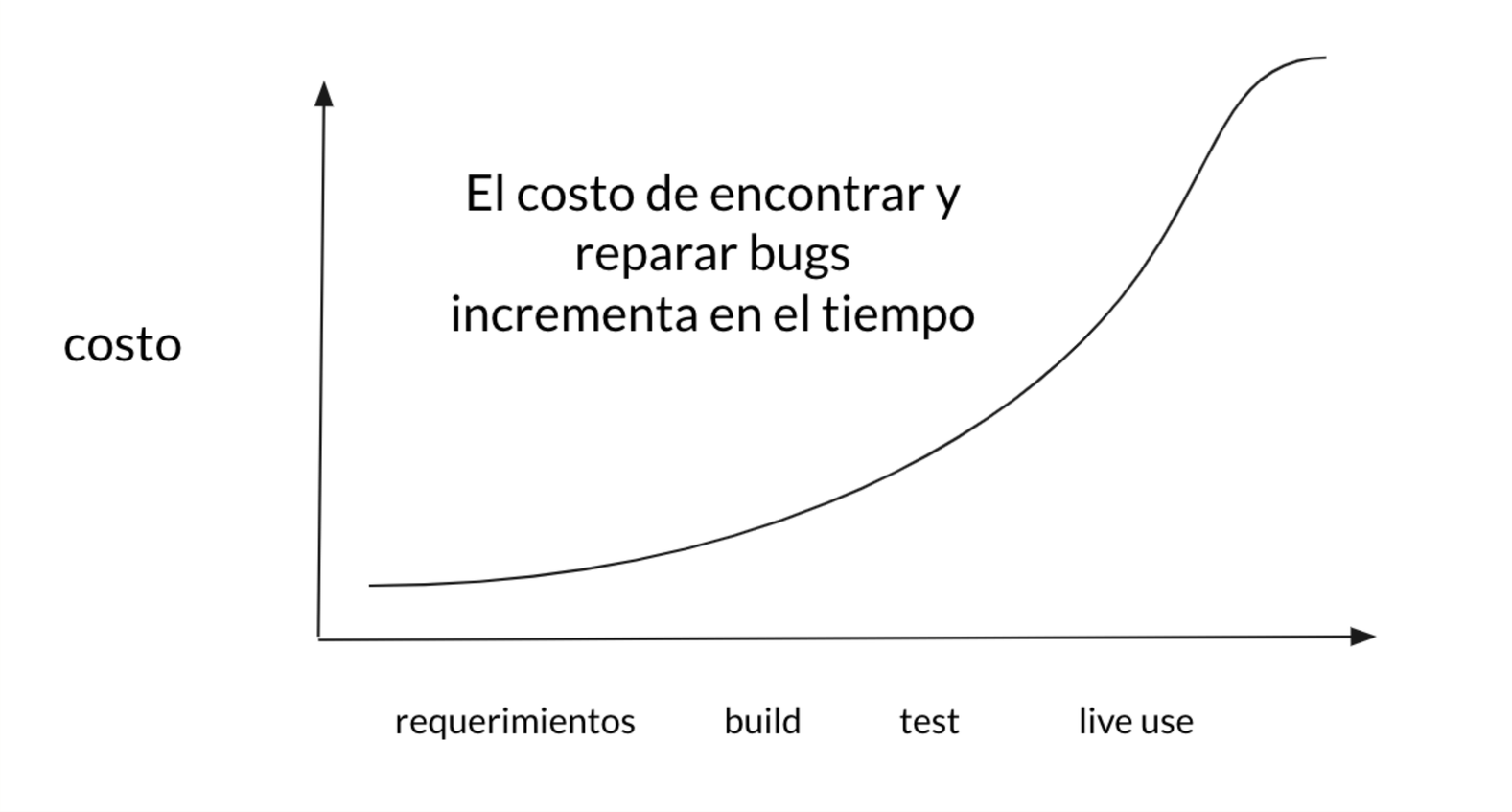
Defecto, Error, o Falla?



Cuando podrían llegar los defectos?



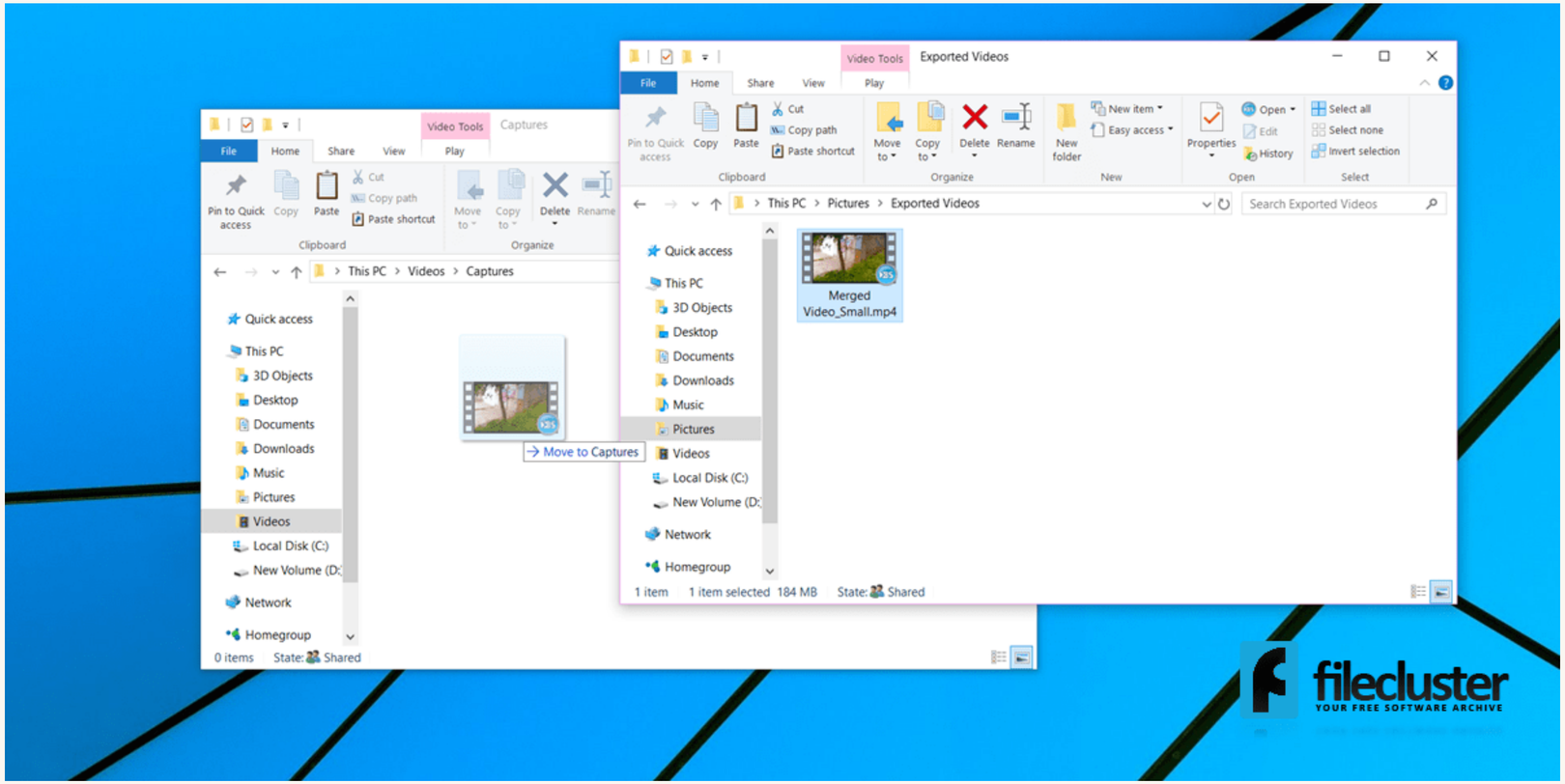
¿Cuál es el costo de un defecto?



Testing

Principios del Testing

¿Cuánto testing es suficiente?



¿Cuánto testing es suficiente?

¿Cuánto testing es suficiente?

Escenarios Posibles

- Tratar de mover un archivo que está abierto
- No tener permisos de escritura en el folder destino
- Disco duro del folder destino está lleno
- Folder destino ya tiene un archivo con ese nombre

Llenar un formulario

si cada formulario puede aceptar
5 posibles tipos de valores.
¿Cuántas combinaciones posibles existirían?

Field 1		Field 6		Field 11	
Field 2		Field 7		Field 12	
Field 3		Field 8		Field 13	
Field 4		Field 9		Field 14	
Field 5		Field 10		Field 15	

Llenar un formulario

si cada formulario puede aceptar
5 posibles tipos de valores.
¿Cuántas combinaciones posibles existirían?

Field 1		Field 6		Field 11	
Field 2		Field 7		Field 12	
Field 3		Field 8		Field 13	
Field 4		Field 9		Field 14	
Field 5		Field 10		Field 15	

5 ^ 15 = *30 517 578 125*

¿Cuánto testing es suficiente?

Principio: Testing exhaustivo es imposible

Probar todas las posibles combinaciones de entradas y precondiciones no es posible, con excepción de los casos triviales. En lugar de hacer un test exhaustivo, se debe analizar los riesgos y priorizar las actividades de testing.

¿Cómo priorizar?

Ejemplo, ¿qué operación es más probable que haga que tu sistema operativo falle?

Abrir microsoft word

Abrir internet explorer

Abrir 10 aplicaciones de gráficas pesadas al mismo tiempo

Ejemplo, ¿qué operación es más probable que haga que tu sistema operativo falle?

Abrir 10 aplicaciones de gráficas pesadas al mismo tiempo

Entonces el módulo más propenso a defectos es el de multi-tasking?

Agrupación de defectos

Principio: Agrupación de defectos

Un pequeño grupo de módulos contienen la mayoría de los defectos descubiertos o muestran la mayor cantidad de fallas operacionales.

Algo paradójico

Principio: Paradoja del pesticida

Si las pruebas se repiten una y otra vez, eventualmente el mismo conjunto de pruebas dejará de encontrar bugs. Para sobrellevar esta paradoja, las pruebas necesitan ser regularmente revisadas. Nuevas y diferentes pruebas necesitan ser escritas para probar diferentes partes del sistema y así poder encontrar una mayor cantidad de potenciales defectos.

Después de un trabajo duro uno puede esperar que el sistema no tenga defectos?

No se puede garantizar la ausencia de defectos

Principio: Testing muestra la presencia de defectos

Testing sólo puede mostrar defectos que están presentes, pero no puede probar la ausencia de los mismos. Testing reduce la probabilidad de que defectos sin descubrir sigan en el software, pero no es una prueba de correctitud.

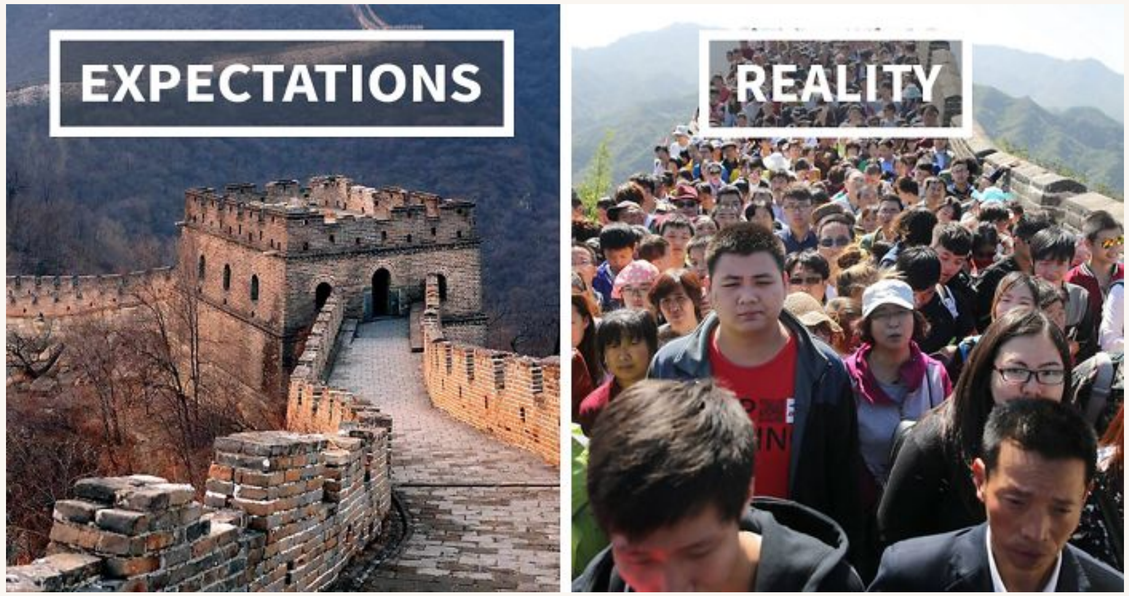
CNN

Bug-free es una falacia

Principio 4 - Falacia de ausencia de errores

Encontrar y arreglar defectos no ayuda si el sistema no es usable y no cumple las necesidades y expectativas del cliente.

Algo a recordar es que si bien corregir defectos es importante, igual de importante es cumplir las necesidades del cliente.



Las pruebas también dependen del contexto

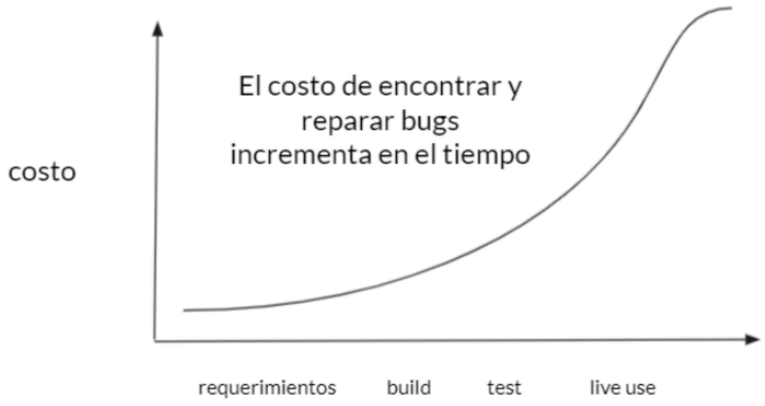
Principio 1 - Testing depende del Contexto.

Testing es realizado diferente en diferentes contextos. Por ejemplo, un programa mas critico es testeado diferente que un sitio de comercio electrónico.

¿Cuándo realizar testing?

Principio 3 - Testing temprano

Las actividades de testing deben empezar tan pronto como sea posible en el ciclo de desarrollo de software.



Lectura Complementaria y Obligatoria

Apuntes

Conceptos Básicos en Canvas