

Apuntes de ingeniería del software

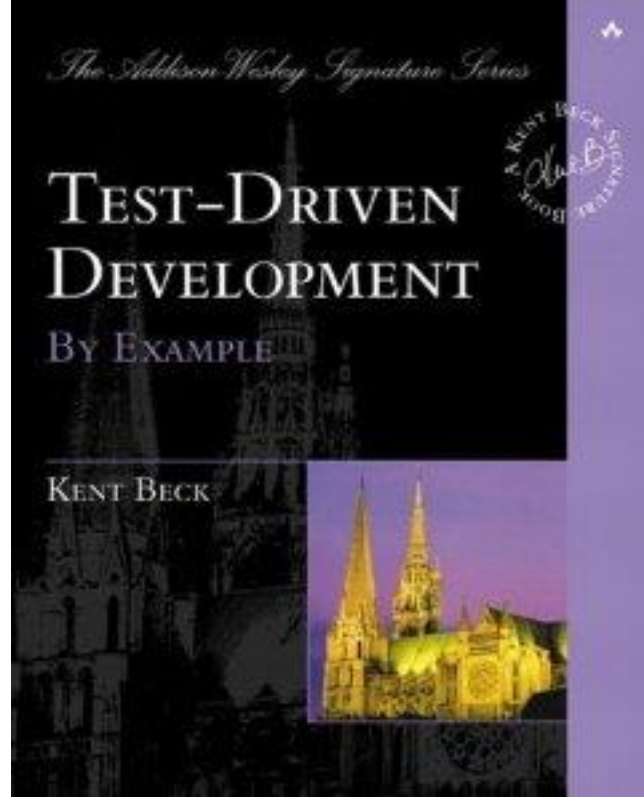
Test Driven Development

José Juan Hernández Cabrera

Escuela Universitaria de Ingeniería Informática. Universidad de
Las Palmas de Gran Canaria
Enero 2014



<http://www.dirigidoportests.com/>



T2. Test Driven Development

Apuntes de ingeniería del software. Escuela Universitaria de Informática. Universidad de Las Palmas de GC
José Juan Hernández Cabrera

¿Para qué sirve?

- Implementar las funciones justas que el cliente necesita y no más.
- Minimizar el número de defectos que llegan al software en fase de producción.
- Modularizar el software para que sea altamente reutilizable y preparado para el cambio.

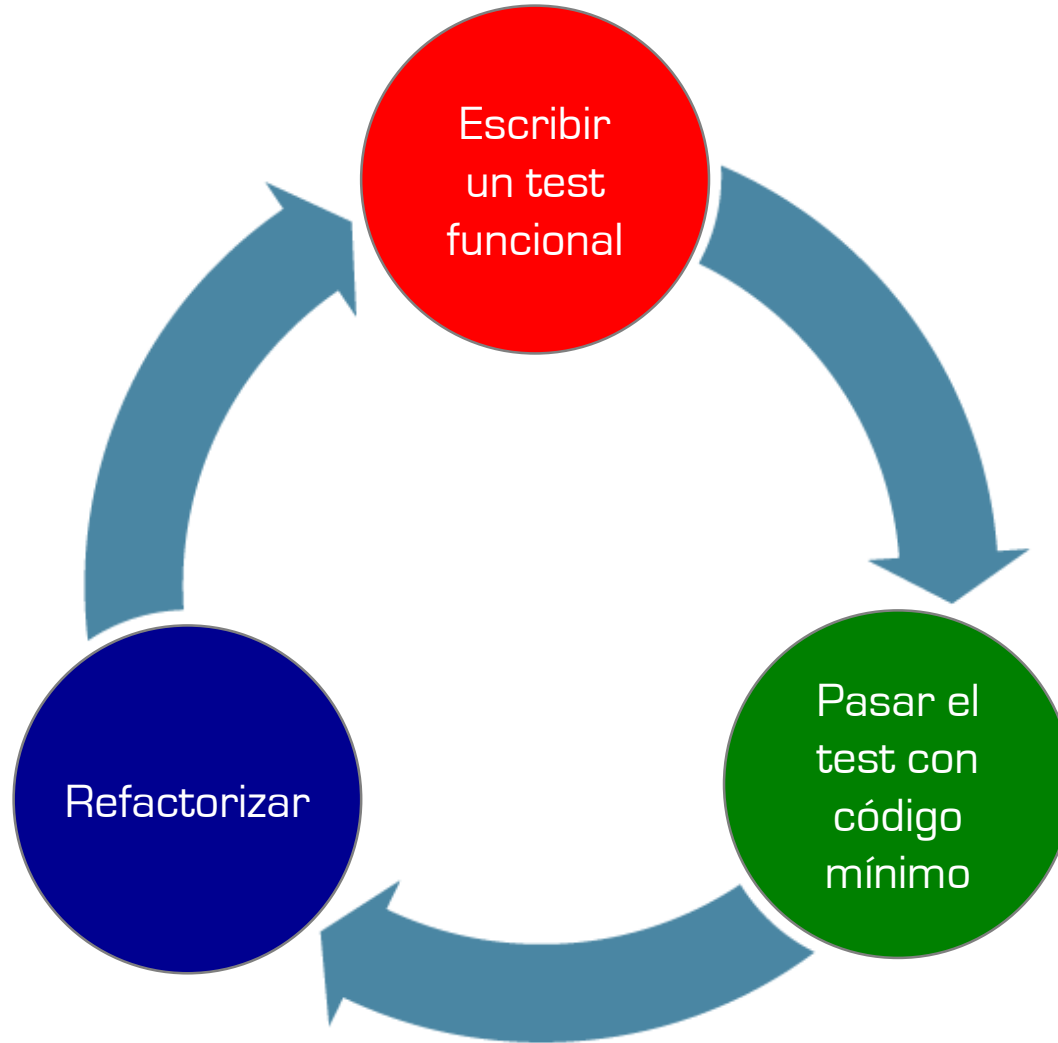
Fundamentos

Realizar el menor diseño posible antes de empezar.

Realizar sólo la Infraestructura del software y el código mínimo para superar los test

Los test guían el diseño.

¿Cómo se hace?



¿Cómo se hace?

Solo se desarrolla funcionalidad cuando exista un test que falle.

No deben existir test que fallen. Por tanto, no se subirán al repositorio código que tenga fallos unitarios.

S.U.T. & D.O.C.

Subject Under Test

Es la entidad que se está probando (sistema, modulo, clase...)

Depended On Component

Es un colaborador, cualquier entidad que requiera el SUT para ejecutar sus tareas.

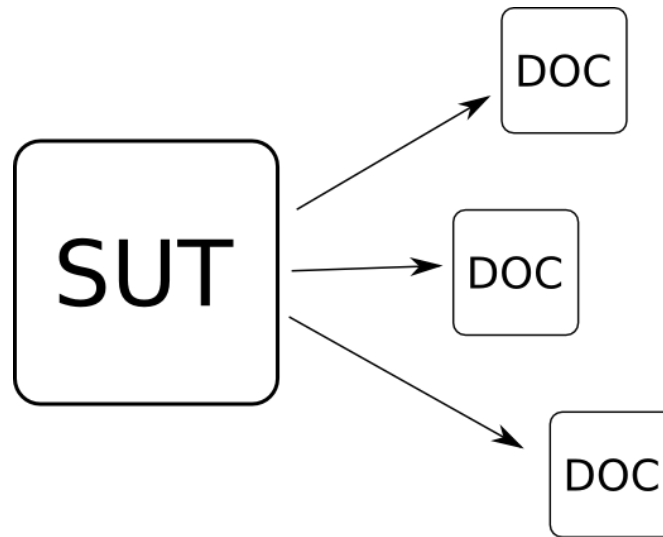
S.U.T. & D.O.C.

SUT es real

DOC puede ser real o simulado

SUT es único

DOC pueden haber muchos



Mocks y Stubs

Los Mocks y Stubs son DOC

El mock se puede definir con mucho detalle y se usa para validar el comportamiento en la colaboración,

El stub sólo devuelve respuestas preprogramadas a posibles llamadas, simplemente simula respuestas a consultas.



Tipos de test

Desarrollo

Test unitario

Test de integración

Test de sistema

Test de rendimiento

Dueño del producto

Test aceptación

Test funcional

Test unitario

Prueba el comportamiento de un módulo aislado de los otros módulos. Es fundamental usar DOC simulados.

Prueba una y solo una característica sin que los demás módulos del software afecten su ejecución.

Atómico
Independiente
Inocuo

Test de integración

Es un test que ayuda a unir distintas partes del sistema.

Pueden escribir y leer de base de datos para comprobar que, efectivamente, la lógica de negocio entiende datos reales.

Test de sistema

Un test del sistema se ejecuta tal cual lo haría el usuario humano, usando los mismos puntos de entrada (interfaz gráfica) y llegando a modificar la base de datos o lo que haya en el otro extremo.

También son conocidos con End-To-End

Test de rendimiento

Carga. Se realiza para observar el comportamiento de una aplicación bajo una cantidad de peticiones esperada, número de usuarios.

Stress. Se realiza para determinar la solidez de la aplicación. El objetivo consiste en identificar en qué condiciones se rompe.

Estabilidad. Se realiza para determinar si la aplicación puede aguantar una carga esperada continuada. Típicamente sirve para identificar “memory leaks”.

Test de aceptación

Es un test que permite comprobar que el software cumple con un requisito de negocio.

Ejemplo

El sistema tiene que estar integrado con la pasarela de pago de Google

El sistema tiene que calcular los indicadores de gestión en menos de 10 minutos.

Test funcional

Un test funcional es un subconjunto de los tests de aceptación. Comprueban alguna funcionalidad con valor de negocio.

Ejemplo

El producto X con precio 50 € tiene un precio final de 55 € después de aplicar el impuesto Z.