Universidad
Francisco de Vitoria
UFV Madrid

Desarrollo e Integración del Software

Tema 3
TDD. Test Driven Development

Grado en Ingeniería Informática
Escuela Politécnica Superior

Test Driven Development

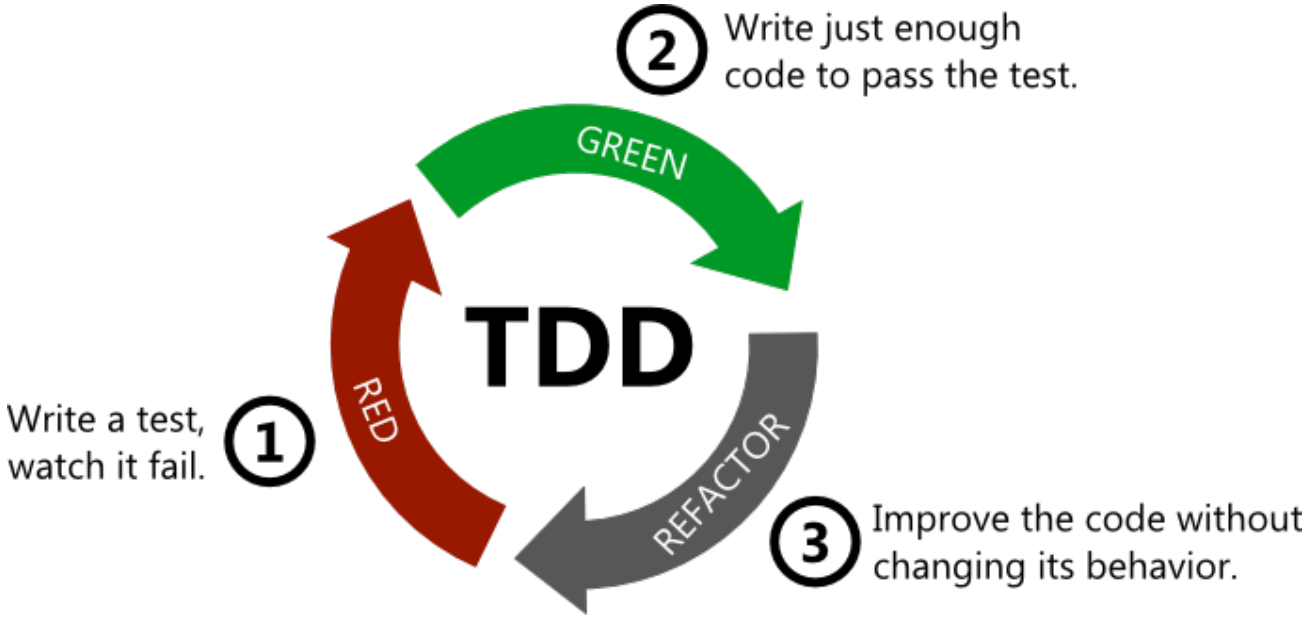
Es una práctica de ingeniería de software que involucra otras dos prácticas:

- Escribir las pruebas primero (Test First Development)
- Refactorización (Refactoring).

La idea es que los requisitos sean traducidos a pruebas, de este modo, cuando las pruebas pasen se garantiza que el software cumple con los requisitos que se han establecido.

Ciclo básico

Universidad Francisco de Vitoria
UFV Madrid



Beneficios de utilizar TDD

Universidad Francisco de Vitoria UFV Madrid

- Minimiza el número de bugs en el código
- Seguridad a la hora de implantar nuevas funcionalidades o realizar cambios en el código. Confianza!
- Feedback instantáneo. Evidencia de que el software funciona.
- Implementar las funcionalidades necesarias y no más. (Evitar el gold-platting, mal llamados “poyaques”)
- Facilita comprobar el entendimiento de los requisitos, y que el software desarrollado los cumple.
- Producir software modular, altamente reutilizable y preparado para el cambio.

Desarrollo con TDD

1. Elegir un requisito
2. Escribir una prueba
3. Verificar que la prueba falla
4. Escribir la implementación
5. Ejecutar las pruebas automatizadas
6. Refactorización
7. Actualización de los requisitos

Limitaciones de TDD

La implementación de TDD no es siempre sencilla, ya que pueden existir limitaciones a la hora de implementar las pruebas unitarias, como, por ejemplo:

- Interfaces Gráfica de usuario (GUIs).
- Objetos distribuidos, aunque los objetos simulados (*MockObjects*) pueden ayudar.
- Bases de datos.

Universidad
Francisco de Vitoria
UFV Madrid

Refactorización

Refactoring

Grado en Ingeniería Informática
Escuela Politécnica Superior

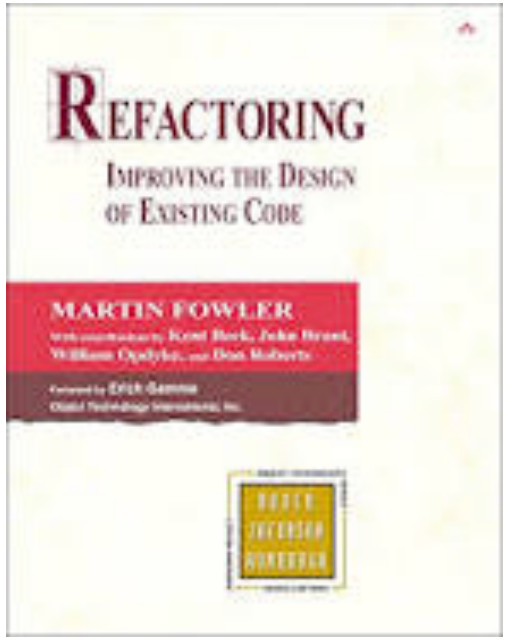
Refactoring

Universidad Francisco de Vitoria UFV Madrid

Definición

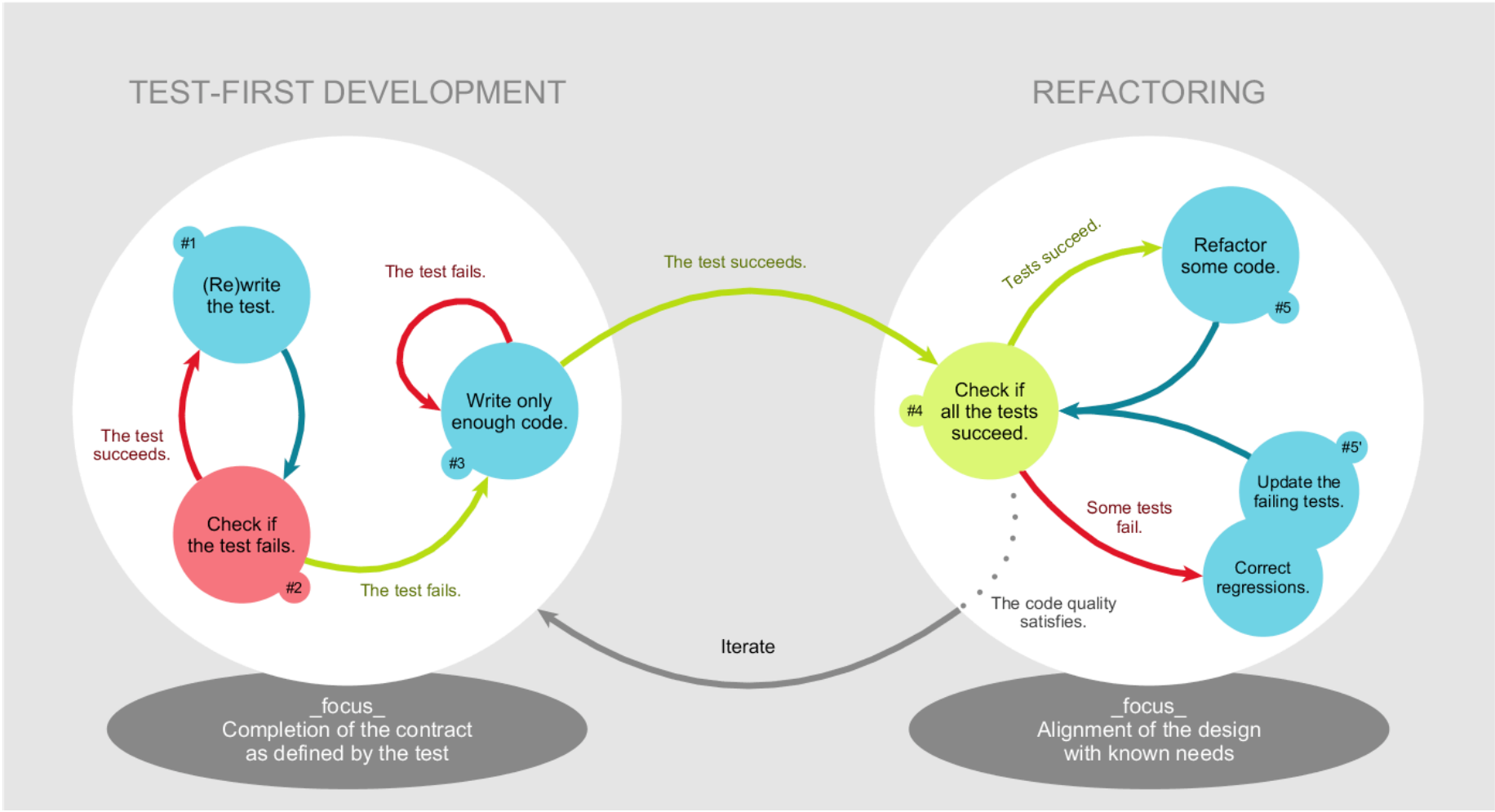
La refactorización (del inglés **refactoring**) es una técnica de la ingeniería de software para reestructurar un código fuente, alterando su estructura interna sin cambiar su comportamiento externo.

- Mejor organizado
- Más limpio
- Optimizarlo, más rápido
- Etc…
- **Multitud de beneficios**



Flujo TDD completo

Universidad Francisco de Vitoria
UFV Madrid



Universidad
Francisco de Vitoria
UFV Madrid

Pruebas unitarias

Grado en Ingeniería Informática
Escuela Politécnica Superior

Pruebas unitarias

Universidad Francisco de Vitoria UFV Madrid

¡No confundir TDD con pruebas unitarias!.

Lo primero es una metodología, mientras que lo segundo es una forma de comprobar que una parte concreta del código funciona correctamente.

TDD hace uso de las pruebas unitarias, pero no son lo mismo.

Pruebas unitarias

Universidad Francisco de Vitoria UFV Madrid

Una prueba unitaria es una forma de comprobar el correcto funcionamiento de una unidad de código. Un algoritmo, un método, una clase, etc…

Además de verificar que el código hace lo que tiene que hacer, indirectamente verificamos otros aspectos del mismo como que los nombres sean correctos, los tipos de los parámetros y los valores que devuelven, etc…

Tipos de pruebas

Universidad Francisco de Vitoria UFV Madrid

- Unitarias
- Integración
- Aceptación
- Aprendizaje. Exploratorias.

Requisitos de las pruebas unitarias

Universidad Francisco de Vitoria UFV Madrid

- **Automatizable:** No debería requerirse una intervención manual. (CI)
- **Completas:** Deben cubrir la mayor cantidad de código posible.
- **Repetibles o Reutilizables:** No se deben crear pruebas que sólo puedan ser ejecutadas una sola vez. (CI)
- **Independientes:** La ejecución de una prueba no debe afectar a la ejecución de otra.
- **Profesionales:** Las pruebas deben ser consideradas igual que el código, con la misma profesionalidad, documentación, etc.

Herramientas Testing

Universidad Francisco de Vitoria UFV Madrid

Existen multitud de herramientas en el mercado. Todos los lenguajes modernos disponen de al menos una.

Algunos ejemplos:

- **Java:** JUnit
- **Javascript**: Jest, Mocha, UnitJS, Jasmine…
- **PHP**: PHPUnit
- **Matlab**: mlUnit
- **LISP**: Five

Lista muy completa de frameworks de testing

Universidad
Francisco de Vitoria
UFV Madrid

JUnit

Grado en Ingeniería Informática
Escuela Politécnica Superior

JUnit

Universidad Francisco de Vitoria UFV Madrid

JUnit es un conjunto de bibliotecas creadas por Erich Gamma y Kent Beck que son utilizadas en programación para hacer pruebas unitarias de aplicaciones Java.

Métodos JUnit

- assertEquals()
- assertNotEquals()
- assertSame()
- assertNull()
- assertTrue()
- assertFalse()
- Fail()
- muchos otros más específicos

Anotaciones en Java

Universidad Francisco de Vitoria UFV Madrid

- Una anotación de Java es una forma de añadir metadatos al código fuente Java, que están disponibles para la aplicación en tiempo de ejecución.
- Muchas veces se usa como una alternativa a la tecnología XML.
- Pueden añadirse a los elementos de programación tales como clases, métodos, metadatos, campos, parámetros, variables locales, y paquetes.
- Las anotaciones pueden incluir parámetros.
- Una anotación muy utilizada en Java es **@Override.**

Anotaciones JUnit

- @Test
  - Expected
  - Timeout
- @ Before
- @ BeforeClass
- @ After
- @ AfterClass
- @ RunWith
  - value
- @Suite
- …

Universidad
Francisco de Vitoria
UFV Madrid

Integración continua

Grado en Ingeniería Informática
Escuela Politécnica Superior

Integración continua

- Propuesto inicialmente por Martin Fowler.
- Consiste en hacer **integraciones** automáticas de un proyecto lo más a menudo posible para así poder detectar fallos cuanto antes.
- Entendemos por integración la **compilación** y ejecución de **pruebas** de todo un proyecto.

- A menudo la integración continua está asociada con las metodologías de programación extrema y desarrollo ágil.

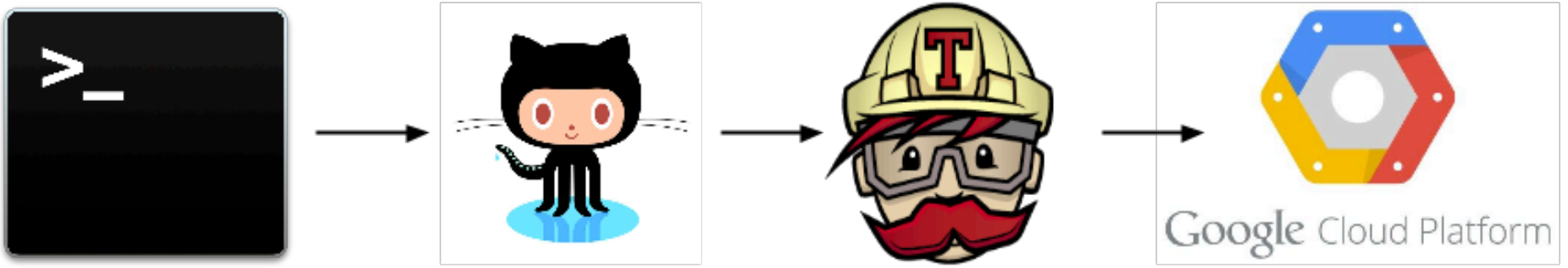
Integración continua

Universidad Francisco de Vitoria UFV Madrid

Jenkins

Travis CI

Atlassian Bamboo



Code

Push to Github

Run local tests on Travis CI

Deploy to Cloud Platform and run end-to-end tests