

UNIT TESTING

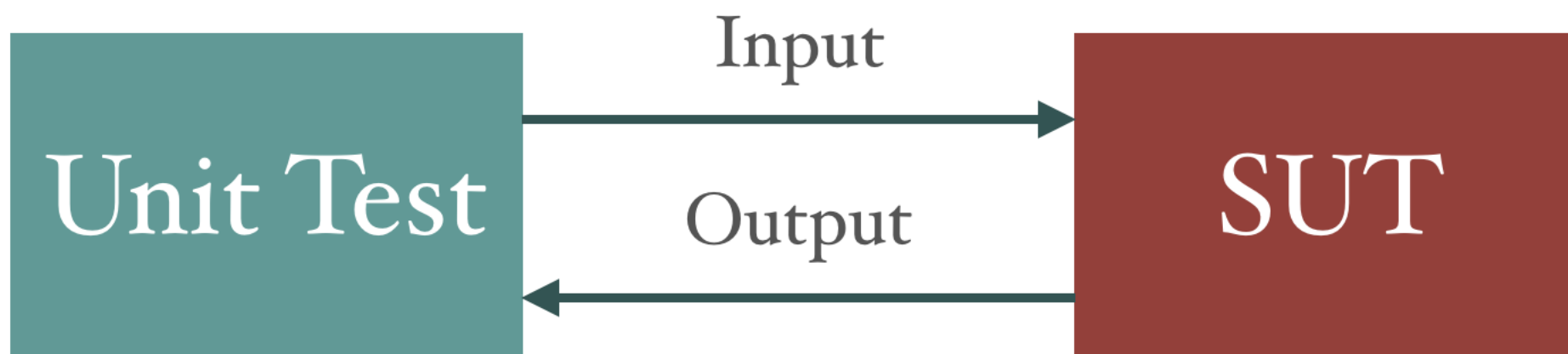
Agenda

- Unit Testing
 - Definición
 - Características
 - Tipos de testeo
 - Nombrado
- TDD
- Coding Dojo

UNIT TESTING (Pruebas Unitarias)

- Método de testeo
- Testea unidades individuales de código
- Unidades como funciones, procedimientos, clases

UNIT TESTING



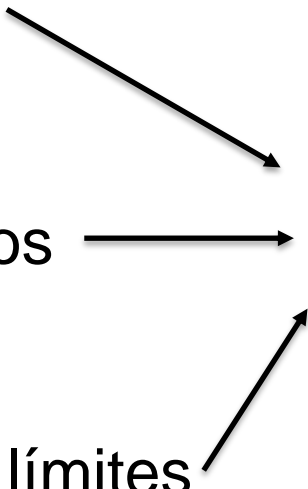
UNIT TESTING

Deberían ser:

- ✓ Automatizados
- ✓ Repetibles
- ✓ Consistentes en resultados
- ✓ Independientes (completamente aislados)
- ✓ Fáciles de implementar
- ✓ De rápida ejecución
- ✓ Legibles
- ✓ Mantenibles

UNIT TESTING

Tipos de testeo:

- Positive testing → inputs validos
 - Negative testing → inputs no validos
 - Borderline testing → inputs son los límites
- comportamiento esperado?
- 
- ```
graph LR; A[Positive testing → inputs validos] --> D[comportamiento esperado?]; B[Negative testing → inputs no validos] --> D; C[Borderline testing → inputs son los límites] --> D;
```

# UNIT TESTING

Tipos de testeo:

- Value-based testing → Verifica valores de retorno
- State-based testing → Verificamos comportamiento/estado del sistema
- Interaction testing → Verificamos mensajes (llamadas) a otros objetos

# UNIT TESTING

## Naming:

- MethodName\_StateUnderTest\_ExpectedBehavior
  - isAdult\_AgeLessThan18\_False
- MethodName\_ExpectedBehavior\_StateUnderTest
  - isAdult\_False\_AgeLessThan18
- Should\_ExpectedBehavior\_When\_StateUnderTest
  - Should\_ThrowException\_When\_AgeLessThan18
- When\_StateUnderTest\_Expect\_ExpectedBehavior
  - When\_AgeLessThan18\_Expect\_isAdultAsFalse



# UNIT TESTING

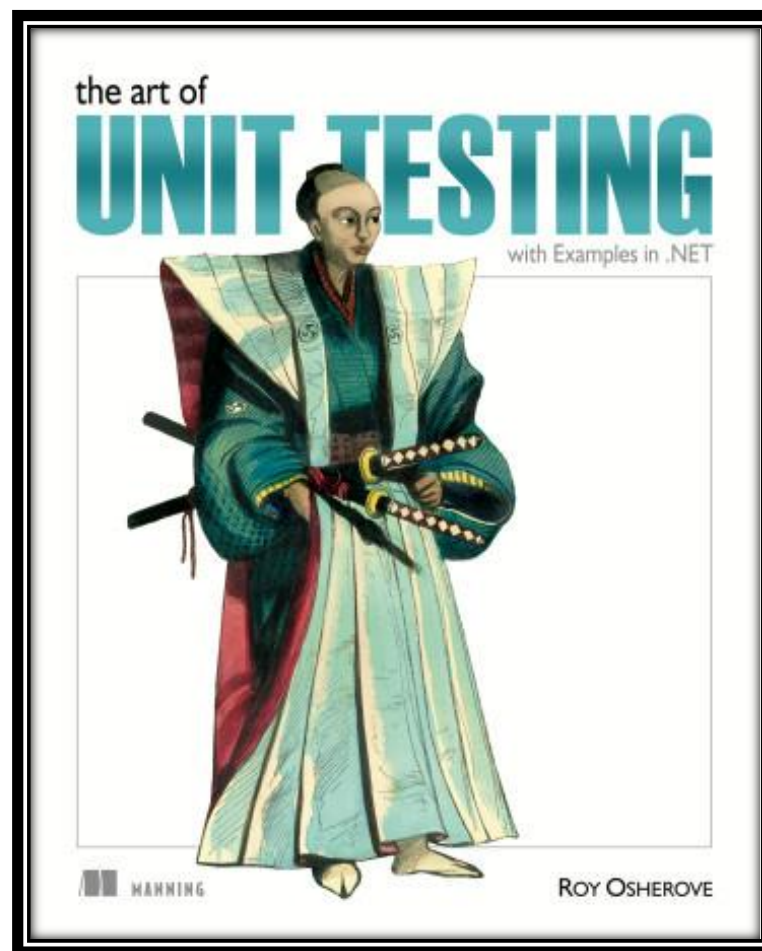
## Unit Testing

Arrange

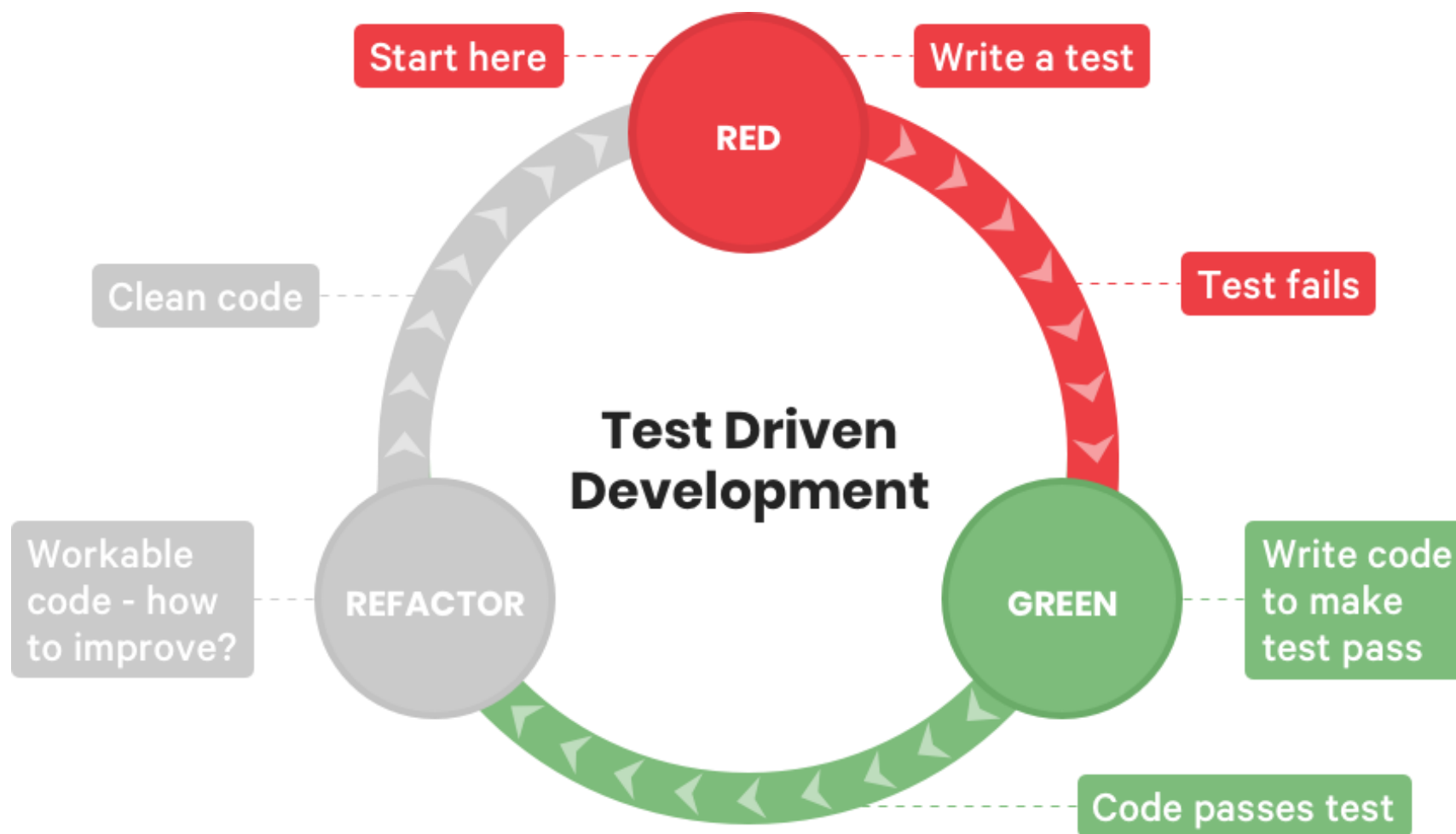
Act

Assert

# UNIT TESTING



# TDD



# TDD

- Diseño evoluciona con la implementación de tests
- Arquitectura y diseño?
- Necesitamos una arquitectura definida/diseñada que establezca límites dentro los cuales se podría aplicar

TDD

TDD

# Test Driven Development is the best thing that has happened to software design



Arek Torczuk

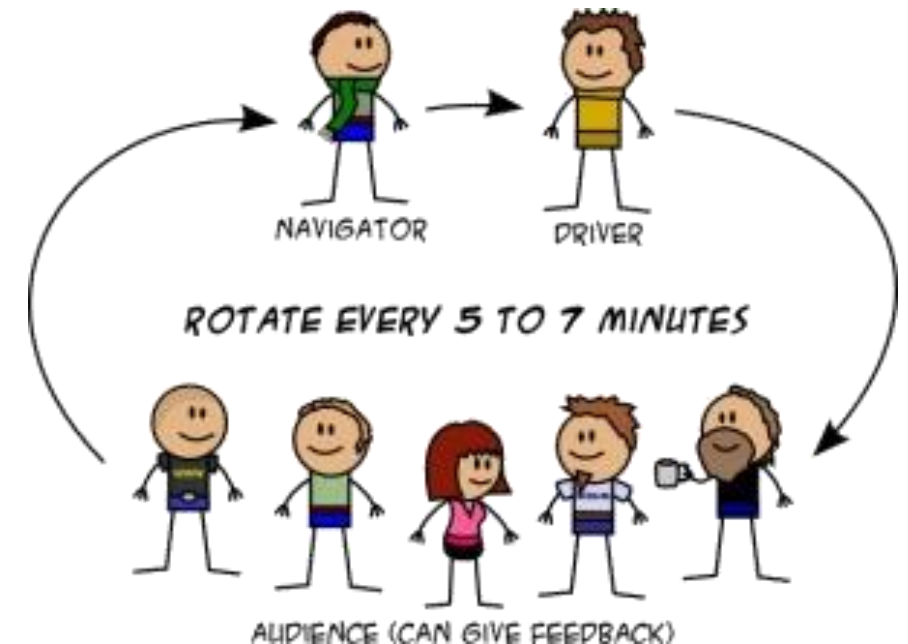
<https://www.thoughtworks.com/insights/blog/test-driven-development-best-thing-has-happened-software-design>

# Coding Dojo

- Es una técnica de entrenamiento conjunto en desarrollo de software
- Sirve y se orienta para aprender de todos, entre todos
- Pueden ser con multiples pair programming que van rotando durante la implementación
- Suele ser abierto al público y con desarrollo cíclico

# Coding Dojo

- Timeboxed
- Comunicación visual
- Promueve varias alternativas de soluciones al problema
- Las practicas o ejercicios son llamadas Katas



# Coding Dojo

## Catálogo de Katas

- KataBankOCR
- FooBarQix
- KataRomanNumerals
- KataRomanCalculator
- KataNumbersInWords
- KataAnagram
- KataDepthFirstSearch