

# TEST DRIVEN DEVELOPMENT

**Introduction, Sample project and more ...**

# PREREQUISITES

- **Basic programming skills (We'll use Java in this workshop)**
- **We'll use spring boot as a vehicle but I'll talk though that as we go**

# INITIAL REQUIREMENTS & SETUP

- **Intellij IDEA**
- **JDK 8 (or JDK9 if you're one of the cool kids)**
- **Repo from Github:**
  - <https://github.com/windwaker/gs-rest-service>
  - git clone ( if you have git installed )
  - Download and extract zip locally
- **Don't wait to ask questions, just fire away**

# INITIAL REQUIREMENTS FOR JDK

www.oracle.com/technetwork/pt/java/javase/downloads/jdk8-downloads-2133151.html

Java EE
Java ME
Java FX
Java DB
Web Tier
Comunidade

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- [Java Developer Newsletter](#): From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
- [Java Developer Day hands-on workshops \(free\)](#) and other events
- [Java Magazine](#)

JDK 8u144 checksum

## Java SE Development Kit 8u144

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

Accept License Agreement  Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.89 MB	<a href="#">jdk-8u144-linux-arm32-vfp-hflt.tar.gz</a>
Linux ARM 64 Hard Float ABI	74.83 MB	<a href="#">jdk-8u144-linux-arm64-vfp-hflt.tar.gz</a>
Linux x86	164.85 MB	<a href="#">jdk-8u144-linux-i586.rpm</a>
Linux x86	179.44 MB	<a href="#">jdk-8u144-linux-i586.tar.gz</a>
Linux x64	162.1 MB	<a href="#">jdk-8u144-linux-x64.rpm</a>
Linux x64	176.92 MB	<a href="#">jdk-8u144-linux-x64.tar.gz</a>
Mac OS X	226.6 MB	<a href="#">jdk-8u144-macosx-x64.dmg</a>
Solaris SPARC 64-bit	139.87 MB	<a href="#">jdk-8u144-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	99.18 MB	<a href="#">jdk-8u144-solaris-sparcv9.tar.gz</a>
Solaris x64	140.51 MB	<a href="#">jdk-8u144-solaris-x64.tar.Z</a>
Solaris x64	96.99 MB	<a href="#">jdk-8u144-solaris-x64.tar.gz</a>
Windows x86	190.94 MB	<a href="#">jdk-8u144-windows-i586.exe</a>
Windows x64	197.78 MB	<a href="#">jdk-8u144-windows-x64.exe</a>

## Java SE Development Kit 8u144 Demos and Samples

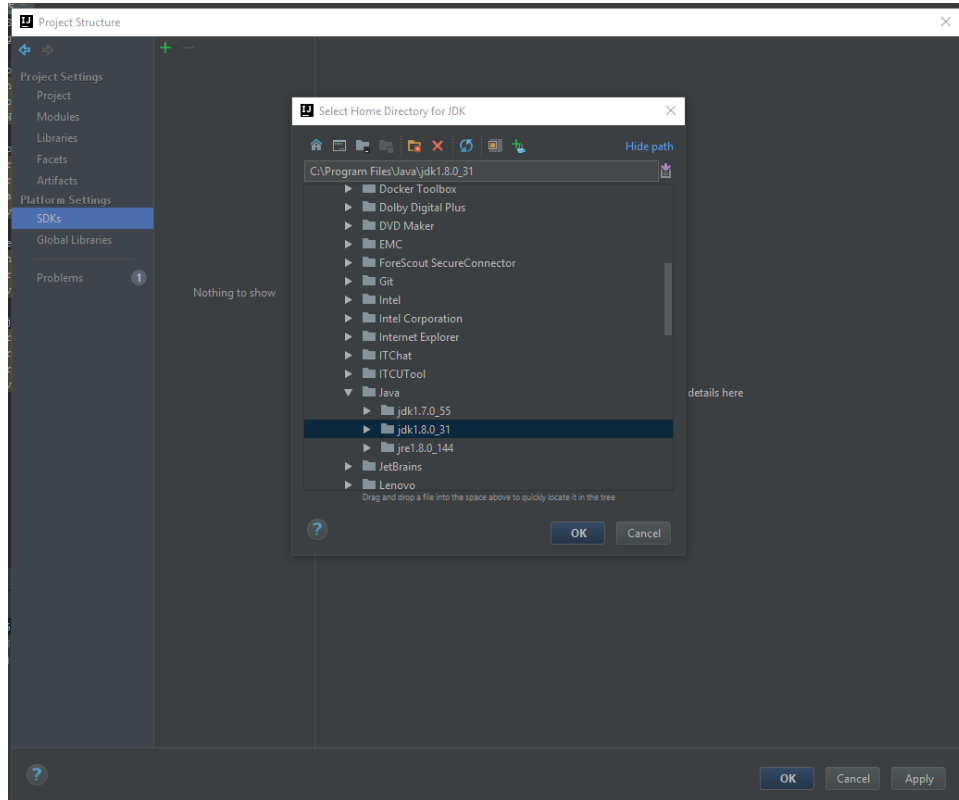
- [Java ME](#)
- [Java Card](#)
- [NetBeans IDE](#)
- [Java Mission Control](#)

### Java Resources

- [Java APIs](#)
- [Technical Articles](#)
- [Demos and Videos](#)
- [Forums](#)
- [Java Magazine](#)
- [Developer Training](#)
- [Tutorials](#)
- [Java.com](#)

- Don't wait to ask questions, just fire away

# INITIAL REQUIREMENTS & SETUP



# INTRODUCTION

- **The Good**
  - Minimise tech debt
  - Allow for rapid refactoring later
  - Safety net for teams

# INTRODUCTION

- **The Bad**
  - Increased complexity initially
  - Need to reset how we write code

# INTRODUCTION

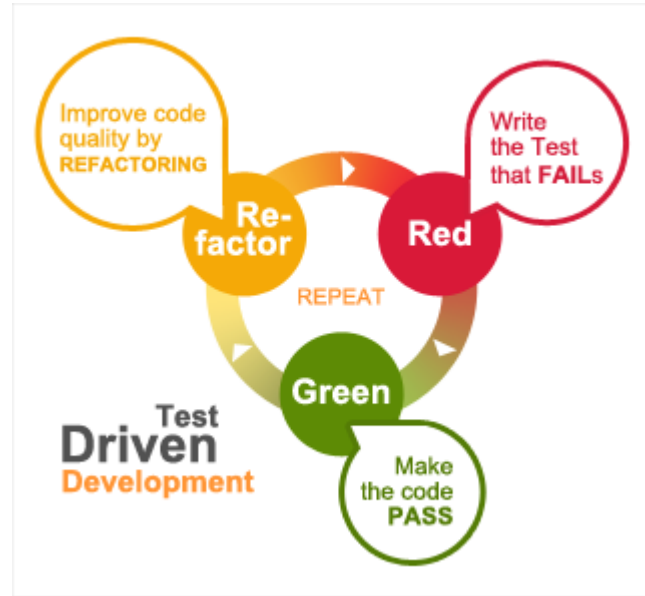
- **The Ugly**





# CHANGE IS COMING ...

TESTING  
IS **NO** LONGER  
TESTING!



**SAMPLE PROJECT**

# SAMPLE REST API

**“Simple REST API with Spring boot”**

**Outside In TDD**

**Write a test to hit rest api (“Write the test you want to read”)**

**Run test & watch it fail**

**Implement endpoints (code in github)**

**Run test and watch it pass**

**Add additional tests**

**Refactor to separate business logic from http**

- Move focus from acceptance (e2e) tests to unit tests
- Easier to debug and quicker to run

# SAMPLE REST API

Listen to the tests, they are talking to you ...

**Optimize for reading failure reports later**

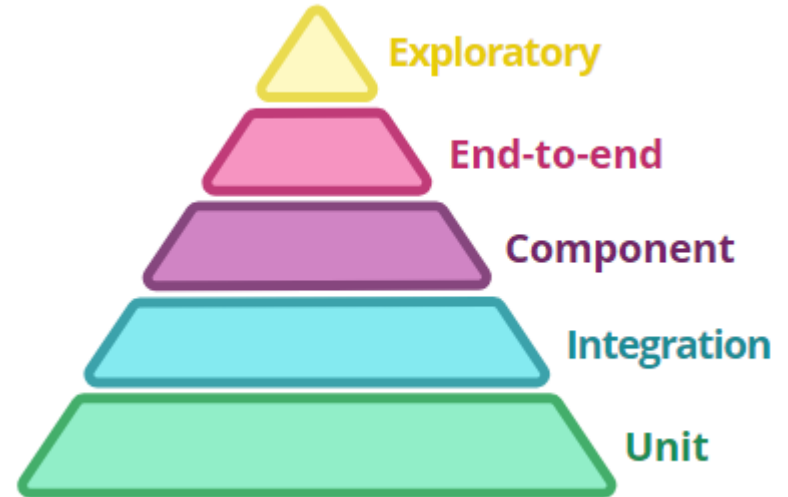
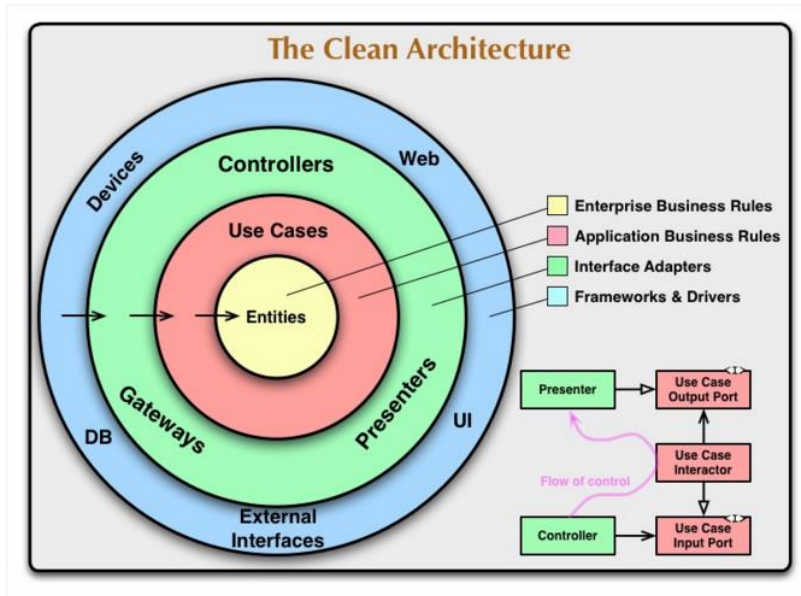
**Some DRY violation is OK in tests (Settle down)**

**Keep tests small and name appropriately**

- Fails for one reason only is the holy grail
- Don't want a big debugging effort when a test fails

**Any time you use the word 'AND', rethink your approach.  
The test is probably doing too much.**

# CLEAN ARCHITECTURE



Clean arch by Uncle Bob, Testing Pyramid by Martin Fowler

# CLEAN ARCHITECTURE

**The outer circles are mechanisms**

- HTTP

**The inner circles are policies**

- Business logic

**Goal is to separate the core logic from the mechanisms of communication**

**Push tests down and replace E2E tests where appropriate.**

**“E2E tests drive the architecture, Unit tests drive the design”**

**– Sagy Rozman**

# SAMPLE REST API

## Code Coverage

**Code Coverage not a cure-all but a good place to start.**

**CODE COVERAGE**



**MOCKING**

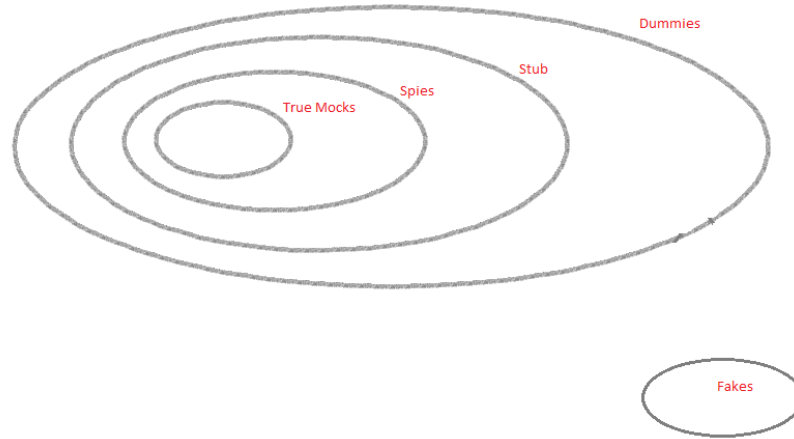


# MOCKING

- **Essential when dealing with complex systems and dependencies**
- **Terms in this area can be very overloaded**

# MOCKING LANDSCAPE

Mocking  
Landscape



# TYPES OF TDD

# TDD IS A DISCIPLINE

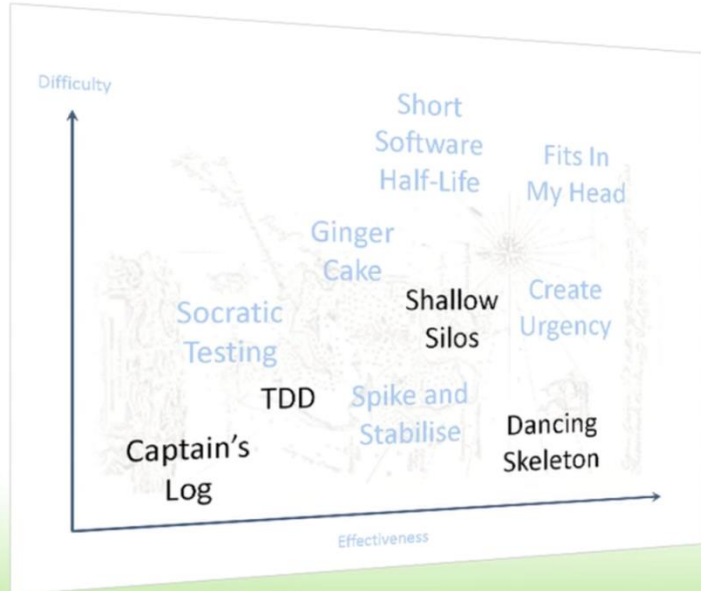
**TDD is not the fastest way to get a task done**

**TDD is the fastest way to manage a code base over time**

**TDD takes time to learn and even longer to see the return**

- (This is where the discipline comes in)

# THE TDD LANDSCAPE



NEXT STEPS

# THE BIGGER PICTURE

- **CI / CD / CT**

- “One lesson that we’ve learned repeatedly is that nothing forces us to understand a process better than trying to automate it.”

Freeman / Pryce

- **The walking skeleton**

- Arms / legs etc can be microservices

# ESSENTIAL READING



# BOOKS

- **Growing Object-Oriented Software, Guided by Tests**
  - Steve Freeman and Nat Pryce
- **Building Microservices (dedicated chapter on testing)**
  - Sam Newman

LINKS

# LINKS

- **Mocking by Uncle Bob**
  - <https://8thlight.com/blog/uncle-bob/2014/05/14/TheLittleMocker.html>
- **Clean Architecture by Uncle Bob**
  - <https://8thlight.com/blog/uncle-bob/2012/08/13/the-clean-architecture.html>
- **Martin Fowler**
  - <https://martinfowler.com/articles/mocksArentStubs.html>

# LINKS CONTINUED

- **Gist for simple web server**

- <https://gist.github.com/windwaker/ce435eeb42fc3541a2f3aec667b6d8cc>

- **Spring Boot initializr**

- <https://start.spring.io/>

- **Excellent TDD presentation on Youtube**

- <https://www.youtube.com/watch?v=zcQOjYXe8vM&t>