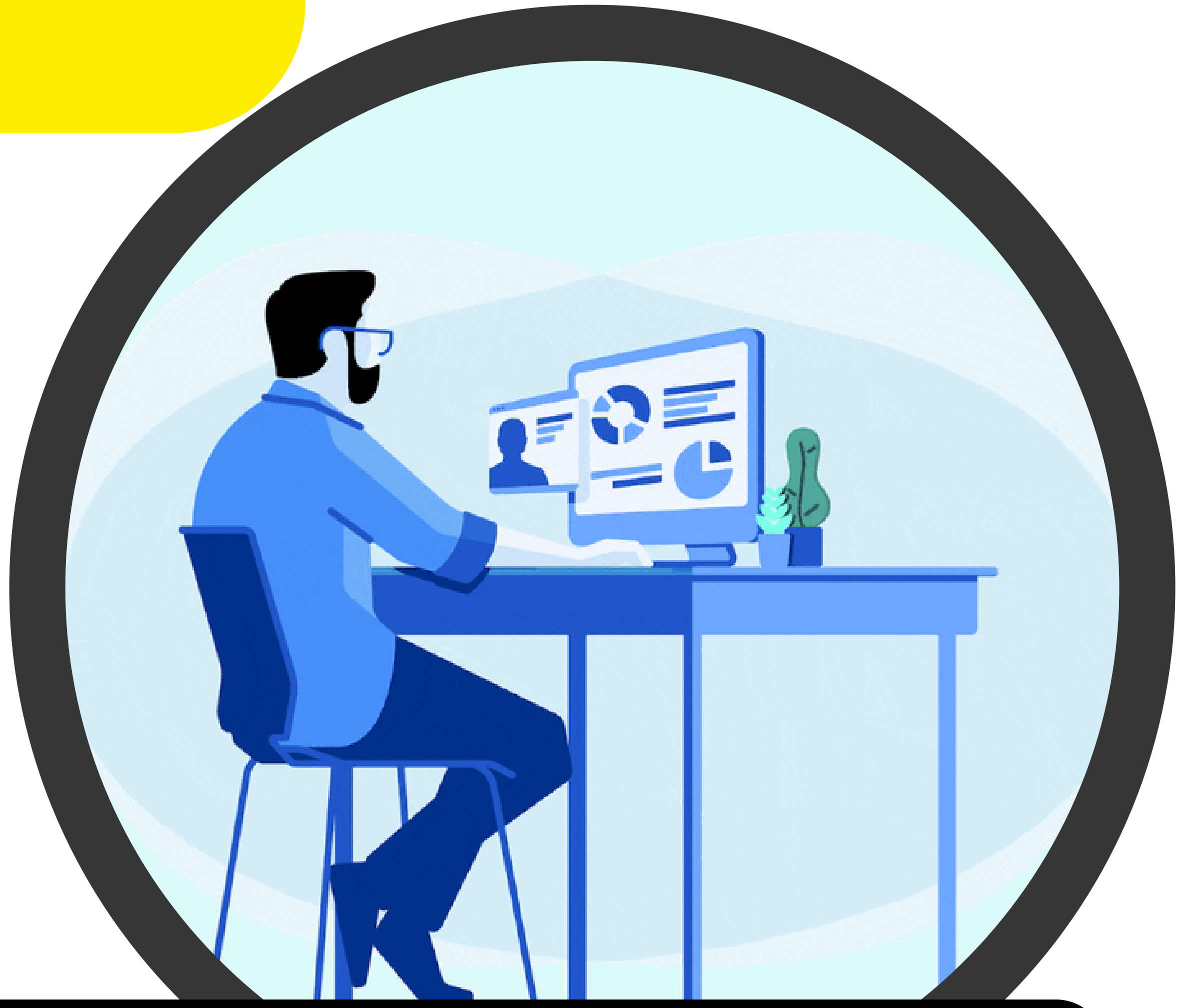
Junit & mockito Introduction

Rajeev Gupta

rgupta.mtech@gmail.com

https://www.linkedin.com/in/rajeev guptajavatrainer



Busy Coder Academy

Agenda

- Introduction to Junit and Mockito
- What is Iterative Software development
- What is TDD?
- Getting started with Junit5 maven configuration
- Junit5 Architecture
- Junit 5 Annotations
- Junit 5 Assertions
- Simple calculator application,Junit 5 annotation examples
- Junit5 how to test method that throws exception
- Junit5 how to test slow Api
- Junit5 how to parameterized test reading from csv file
- Junit5 how to write test suite
- Junit5 and Mockito integration

Busy Coder Academy

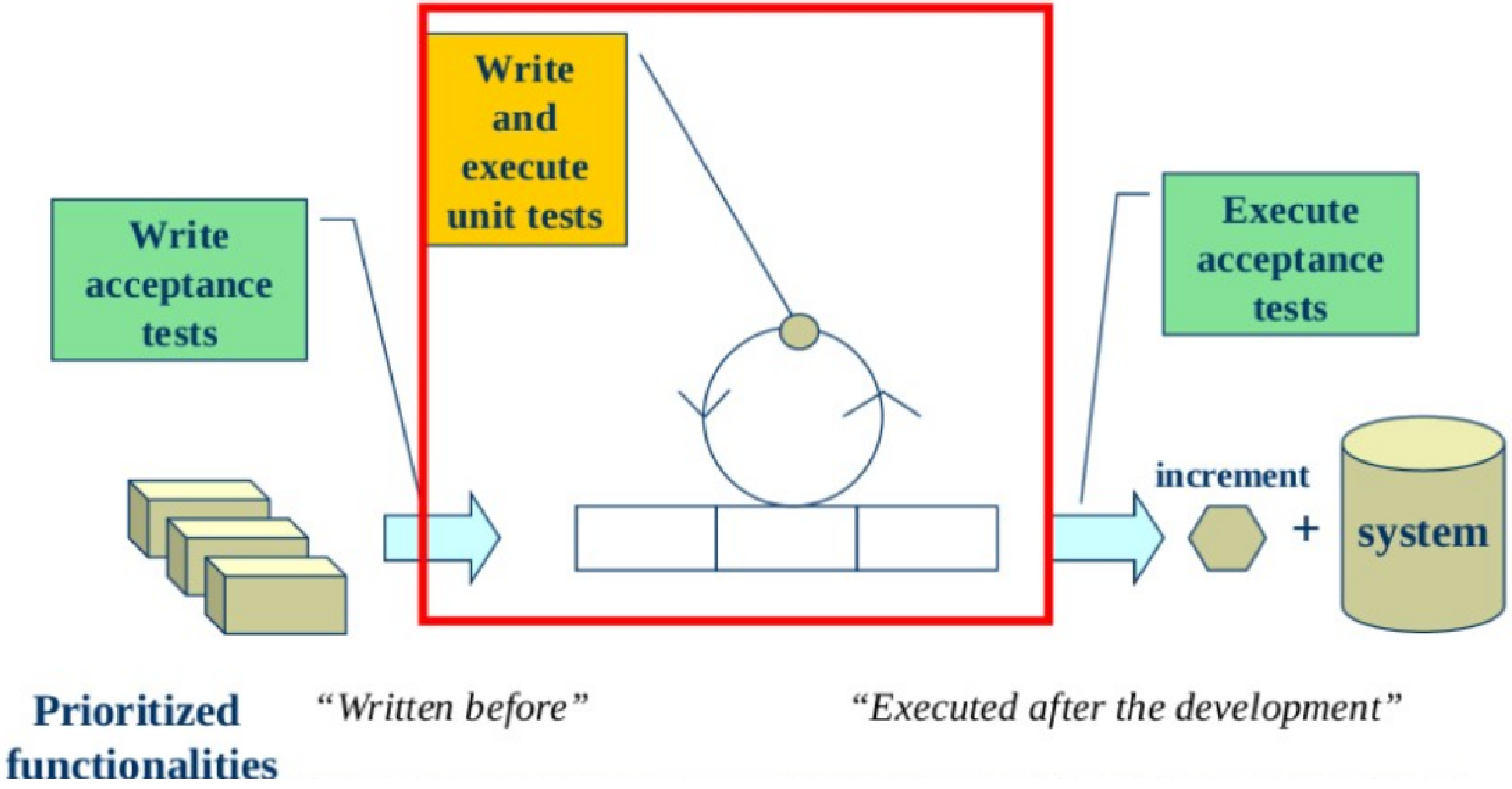
What is Junit?

JUnit is a unit testing open-source framework for the Java programming language. Java Developers use this framework to write and execute automated tests. In Java, there are test cases that have to be re-executed every time a new code is added. This is done to make sure that nothing in the code is broken.

What is Mockito?

Mockito is a mocking framework, JAVA-based library that is used for effective unit testing of JAVA applications. Mockito is used to mock interfaces so that a dummy functionality can be added to a mock interface that can be used in unit testing.

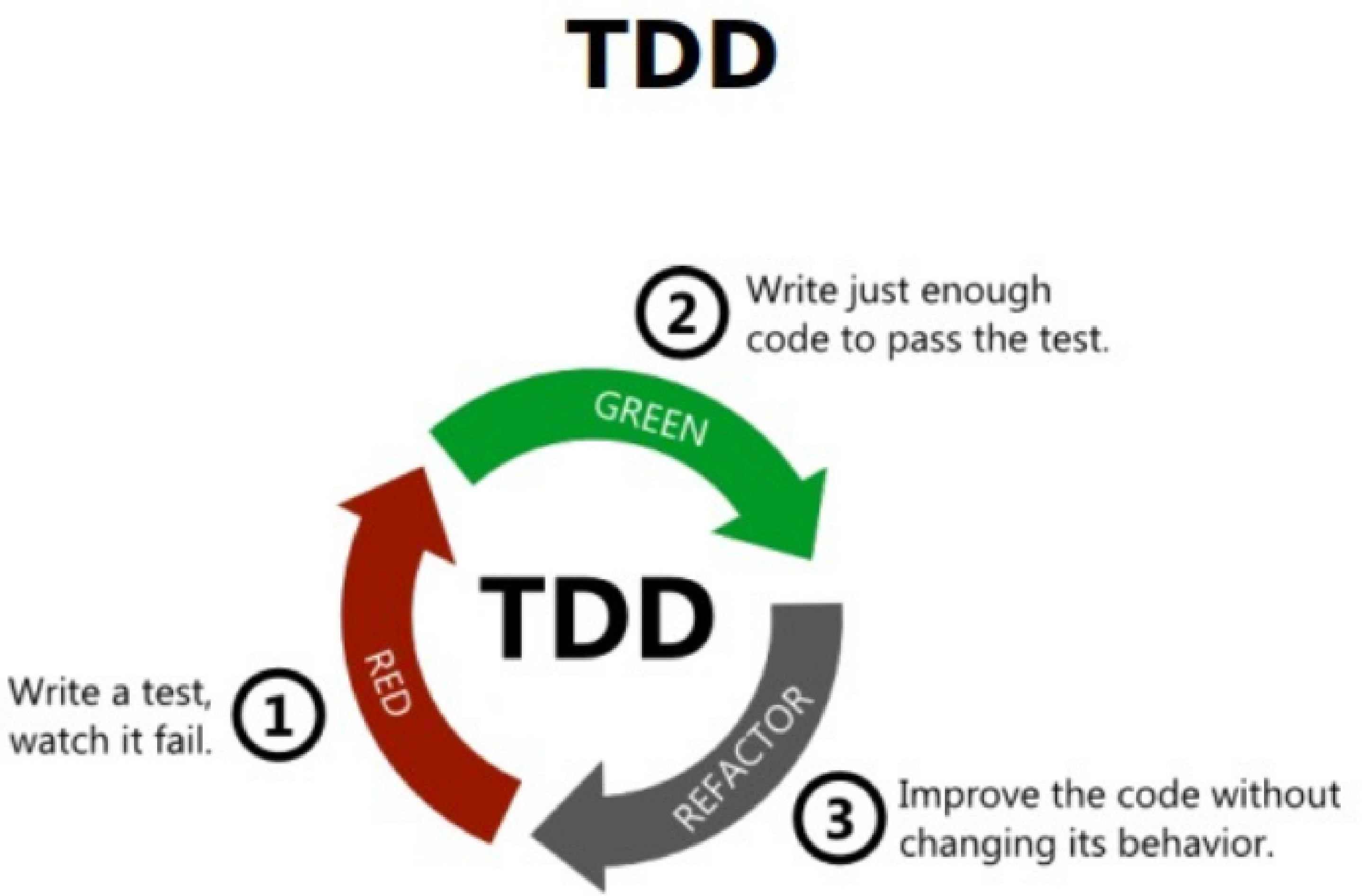
Iterative software development


Write and execute unit tests
Write acceptance tests
Execute acceptance tests
increment
+
system
Prioritized functionalities
“Written before”
“Executed after the development”

Iterative development is a way of breaking down the software development of a large application into smaller chunks. In iterative development, feature code is designed, developed and tested in repeated cycles.

Busy Coder Academy

What is TDD?



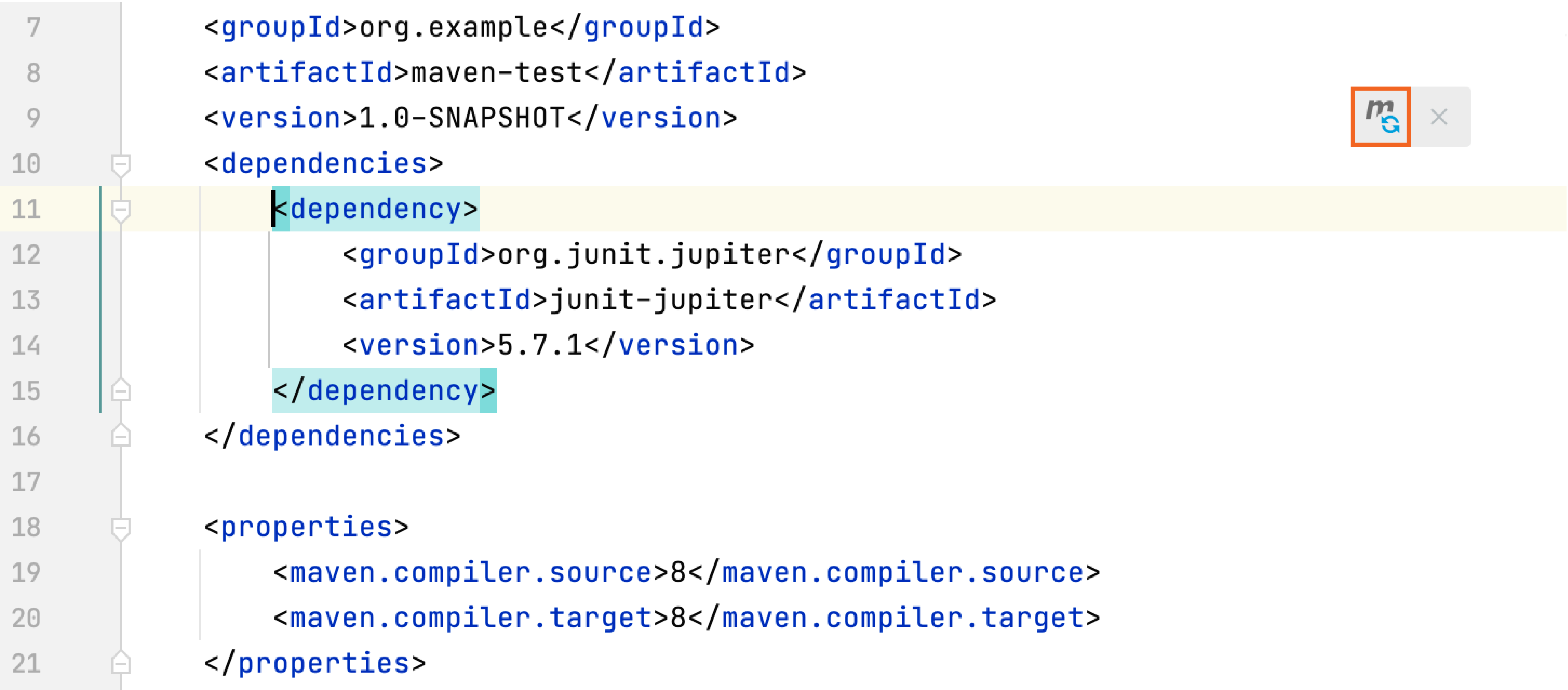
1 Write a test, watch it fail.

2 Write just enough code to pass the test.

3 Improve the code without changing its behavior.

It is an iterative approach combining programming, unit test creation, and refactoring. The TDD approach originates from the Agile manifesto is TDD?

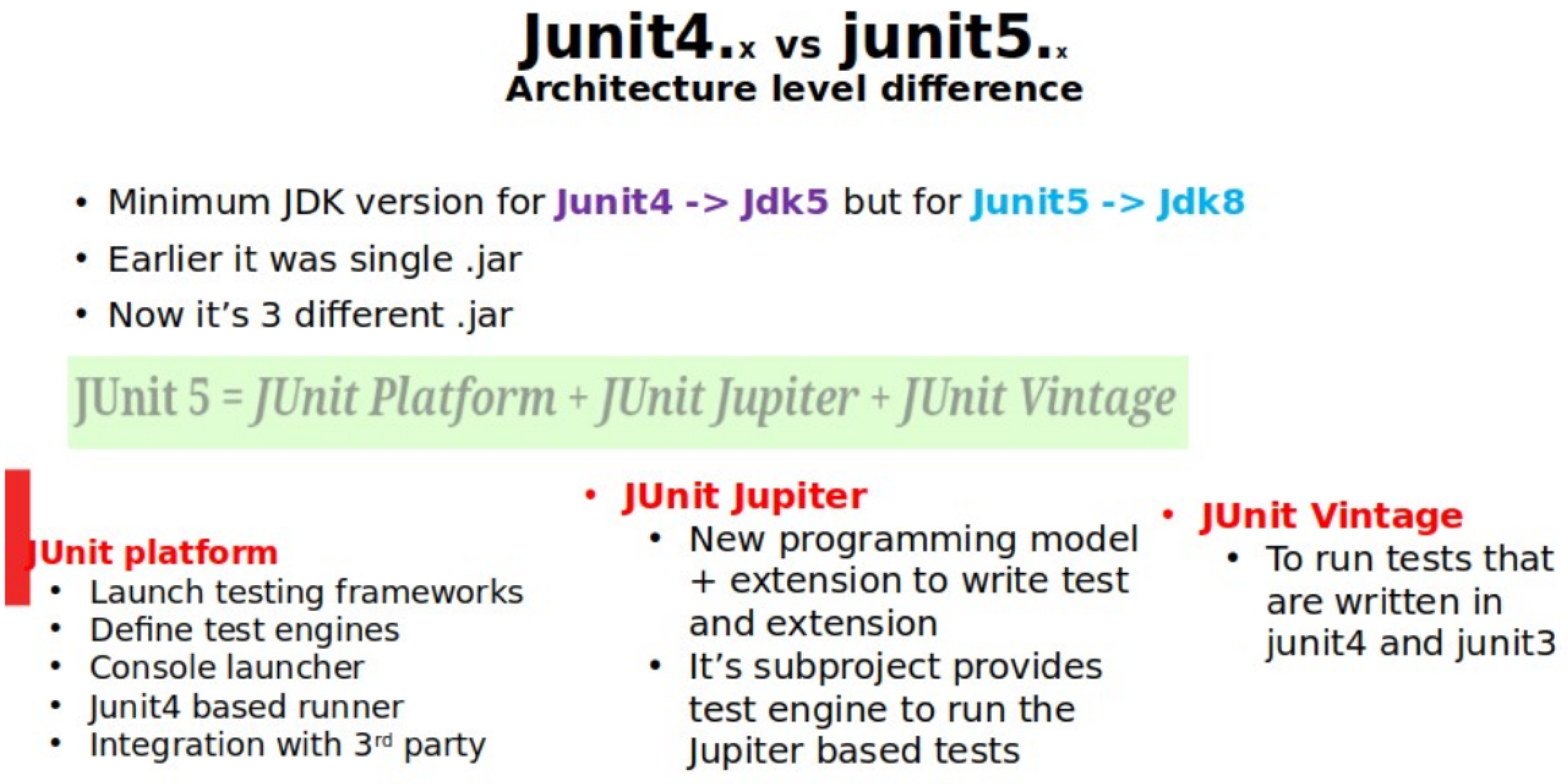
junit 5 maven configuration



```
    <groupId>org.example</groupId>
    <artifactId>maven-test</artifactId>
    <version>1.0-SNAPSHOT</version>
    <dependencies>
        <dependency>
            <groupId>org.junit.jupiter</groupId>
            <artifactId>junit-jupiter</artifactId>
            <version>5.7.1</version>
        </dependency>
    </dependencies>

    <properties>
        <maven.compiler.source>8</maven.compiler.source>
        <maven.compiler.target>8</maven.compiler.target>
    </properties>
```

Junit 4 vs Junit 5



Junit4.$_x$ vs junit5.$_x$

Architecture level difference

- Minimum JDK version for **Junit4 -> Jdk5** but for **Junit5 -> Jdk8**
- Earlier it was single .jar
- Now it's 3 different .jar

JUnit 5 = JUnit Platform + JUnit Jupiter + JUnit Vintage

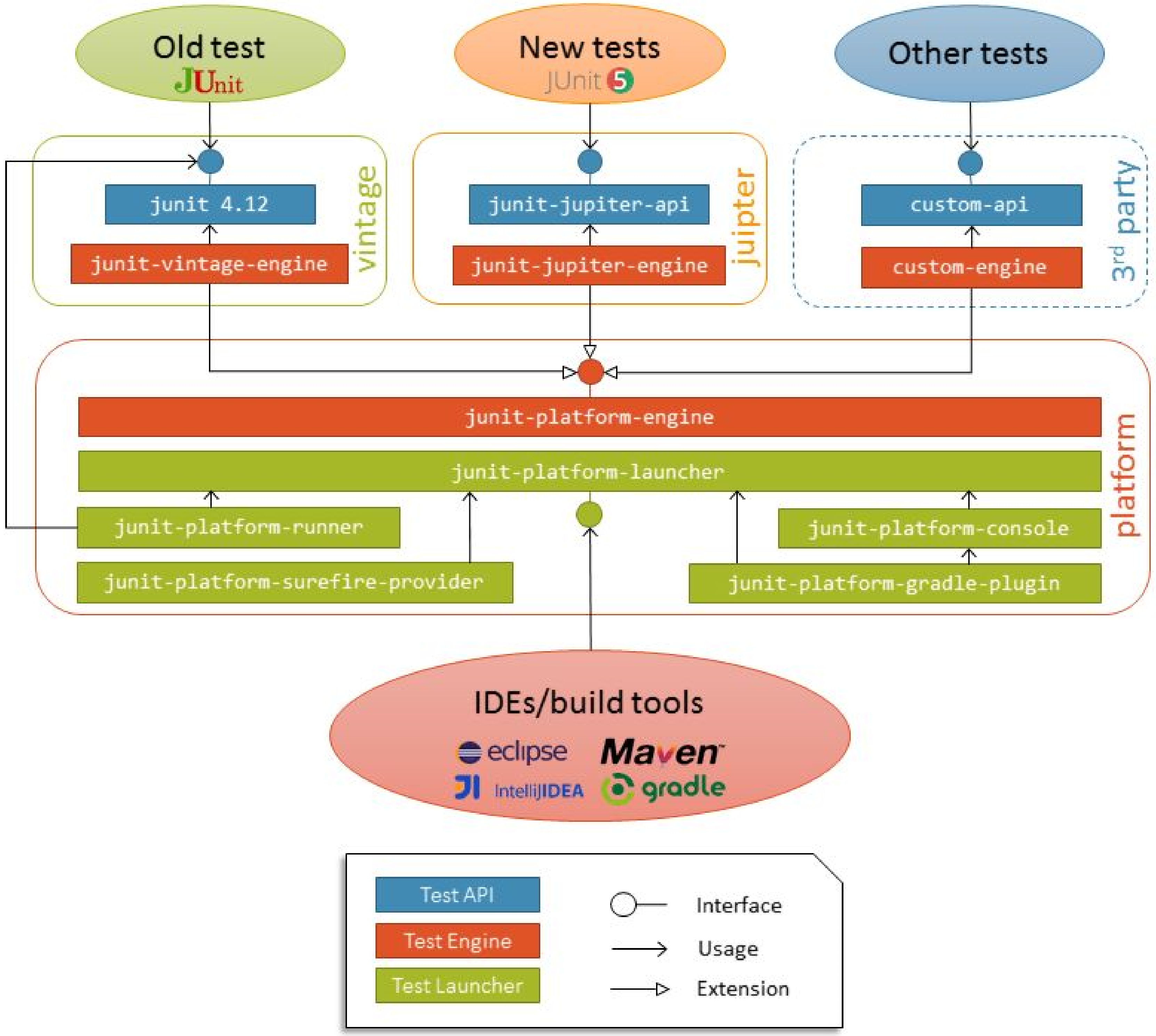
JUnit platform

- Launch testing frameworks
- Define test engines
- Console launcher
- Junit4 based runner
- Integration with 3rd party

- **JUnit Jupiter**
  - New programming model + extension to write test and extension
  - It's subproject provides test engine to run the Jupiter based tests

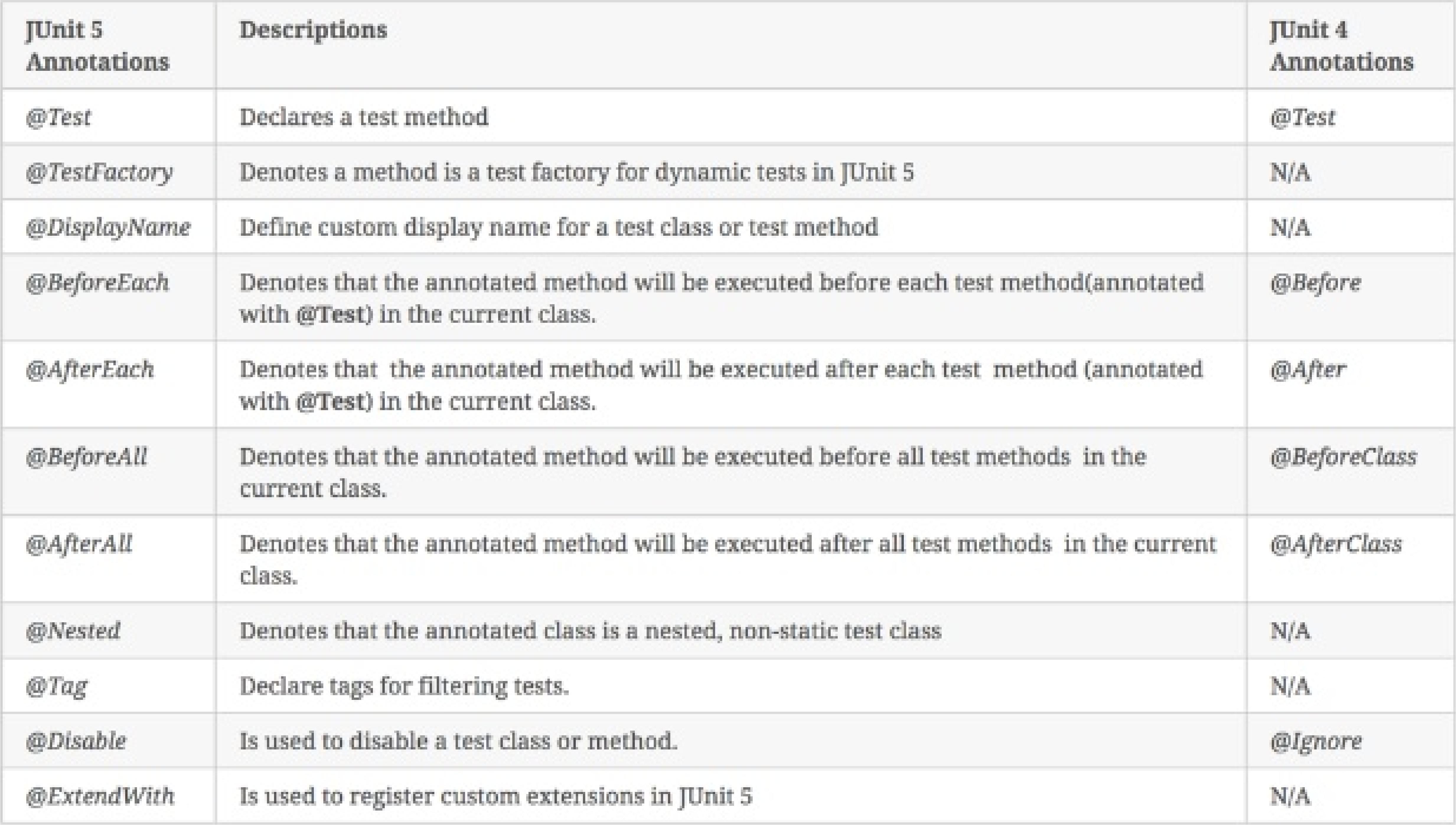
- **JUnit Vintage**
  - To run tests that are written in junit4 and junit3

Junit 5 Architecture



Busy Coder Academy

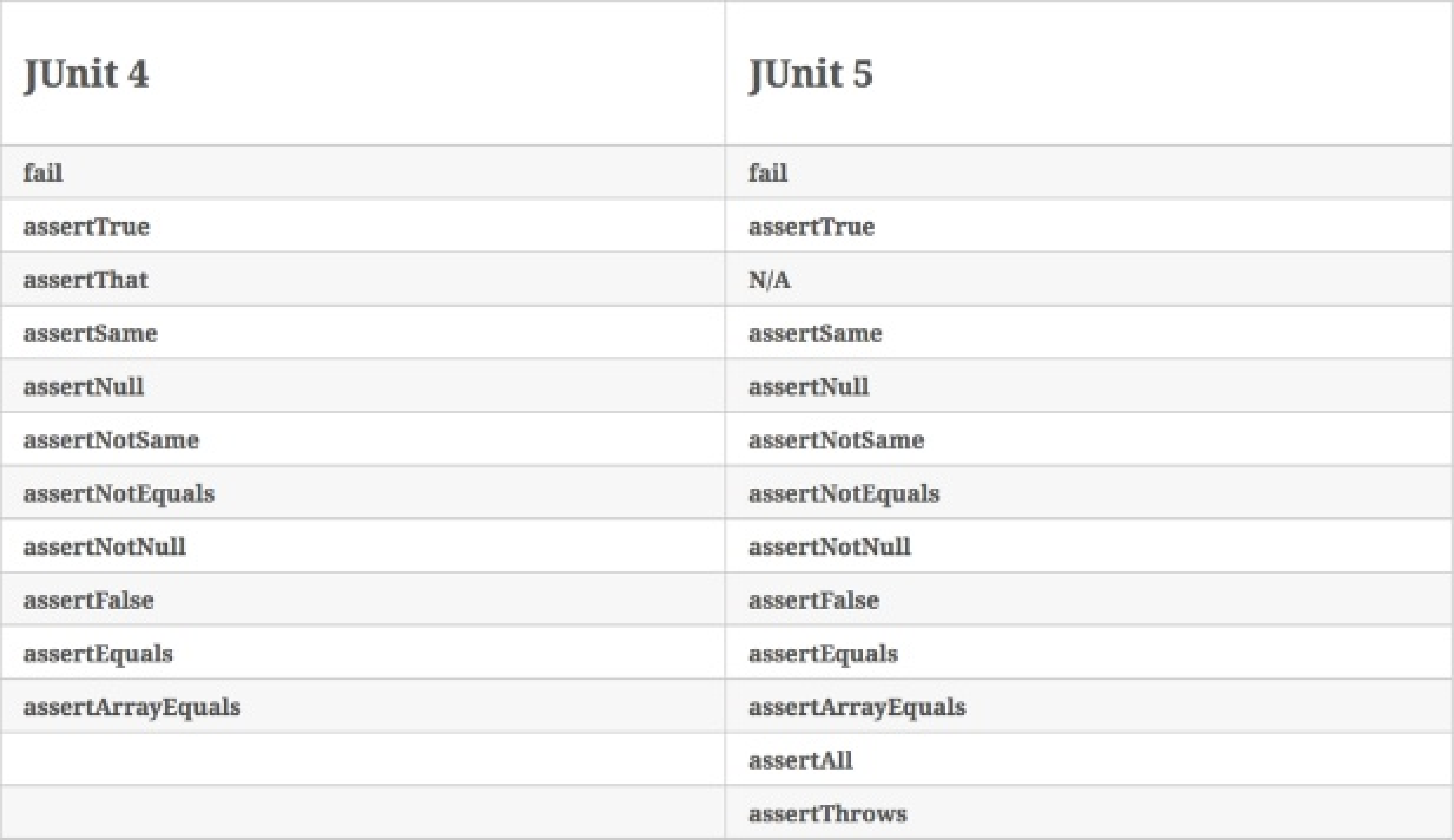
Junit 5 Annotations



JUnit 5 Annotations	Descriptions	JUnit 4 Annotations
@Test	Declares a test method	*@Test*
@TestFactory	Denotes a method is a test factory for dynamic tests in JUnit 5	N/A
@DisplayName	Define custom display name for a test class or test method	N/A
@BeforeEach	Denotes that the annotated method will be executed before each test method(annotated with **@Test**) in the current class.	*@Before*
@AfterEach	Denotes that the annotated method will be executed after each test method (annotated with **@Test**) in the current class.	*@After*
@BeforeAll	Denotes that the annotated method will be executed before all test methods in the current class.	*@BeforeClass*
@AfterAll	Denotes that the annotated method will be executed after all test methods in the current class.	*@AfterClass*
@Nested	Denotes that the annotated class is a nested, non-static test class	N/A
@Tag	Declare tags for filtering tests.	N/A
@Disable	Is used to disable a test class or method.	*@Ignore*
@ExtendWith	Is used to register custom extensions in JUnit 5	N/A

Busy Coder Academy

Junit 5 Assertions



JUnit 4	JUnit 5
fail	fail
assertTrue	assertTrue
assertThat	N/A
assertSame	assertSame
assertNull	assertNull
assertNotSame	assertNotSame
assertNotEquals	assertNotEquals
assertNotNull	assertNotNull
assertFalse	assertFalse
assertEquals	assertEquals
assertArrayEquals	assertArrayEquals
	assertAll
	assertThrows

Junit Mockito Integration

Mockito is a mocking framework for Java. Mockito allows convenient creation of substitutes of real objects for testing purposes.