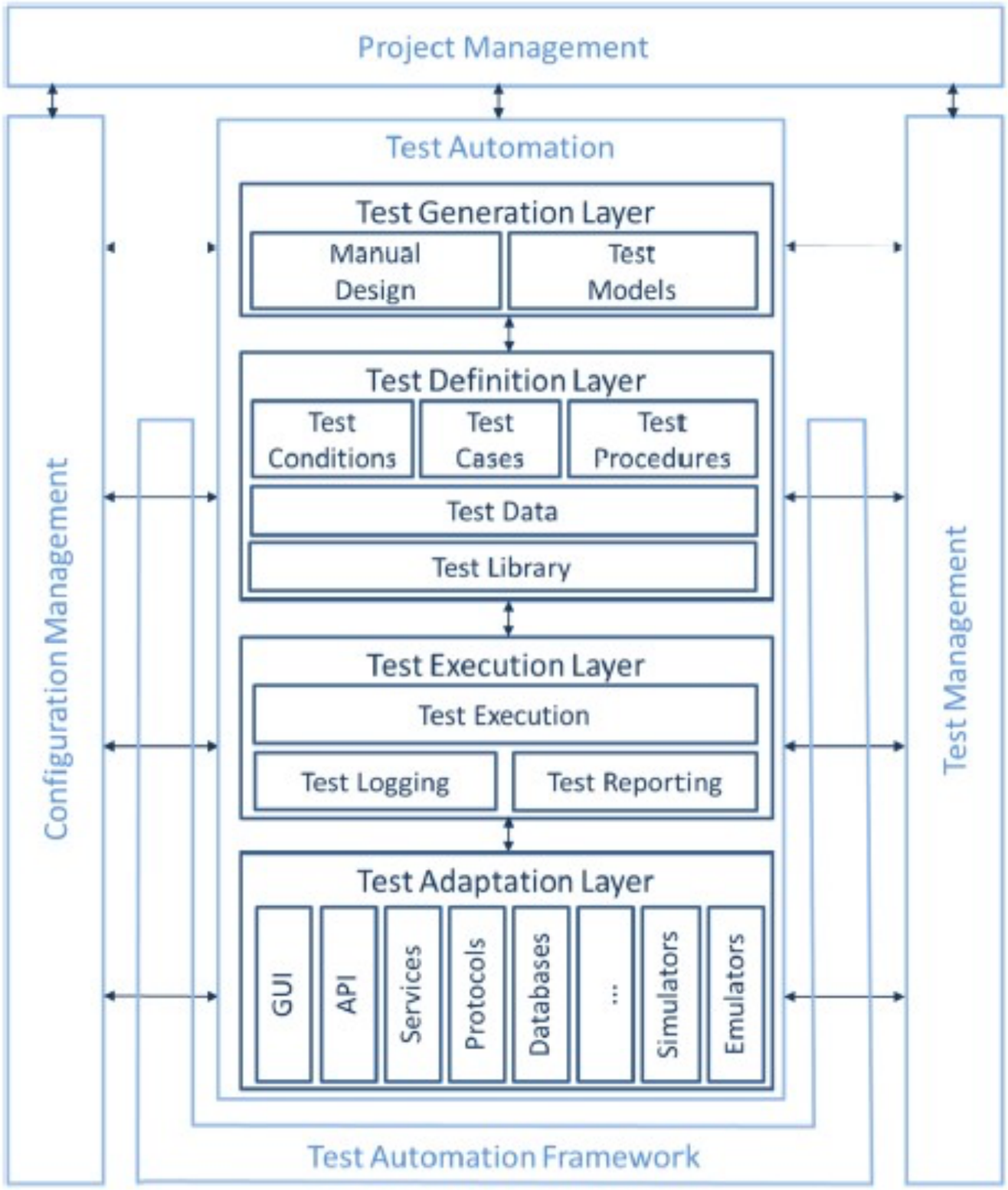
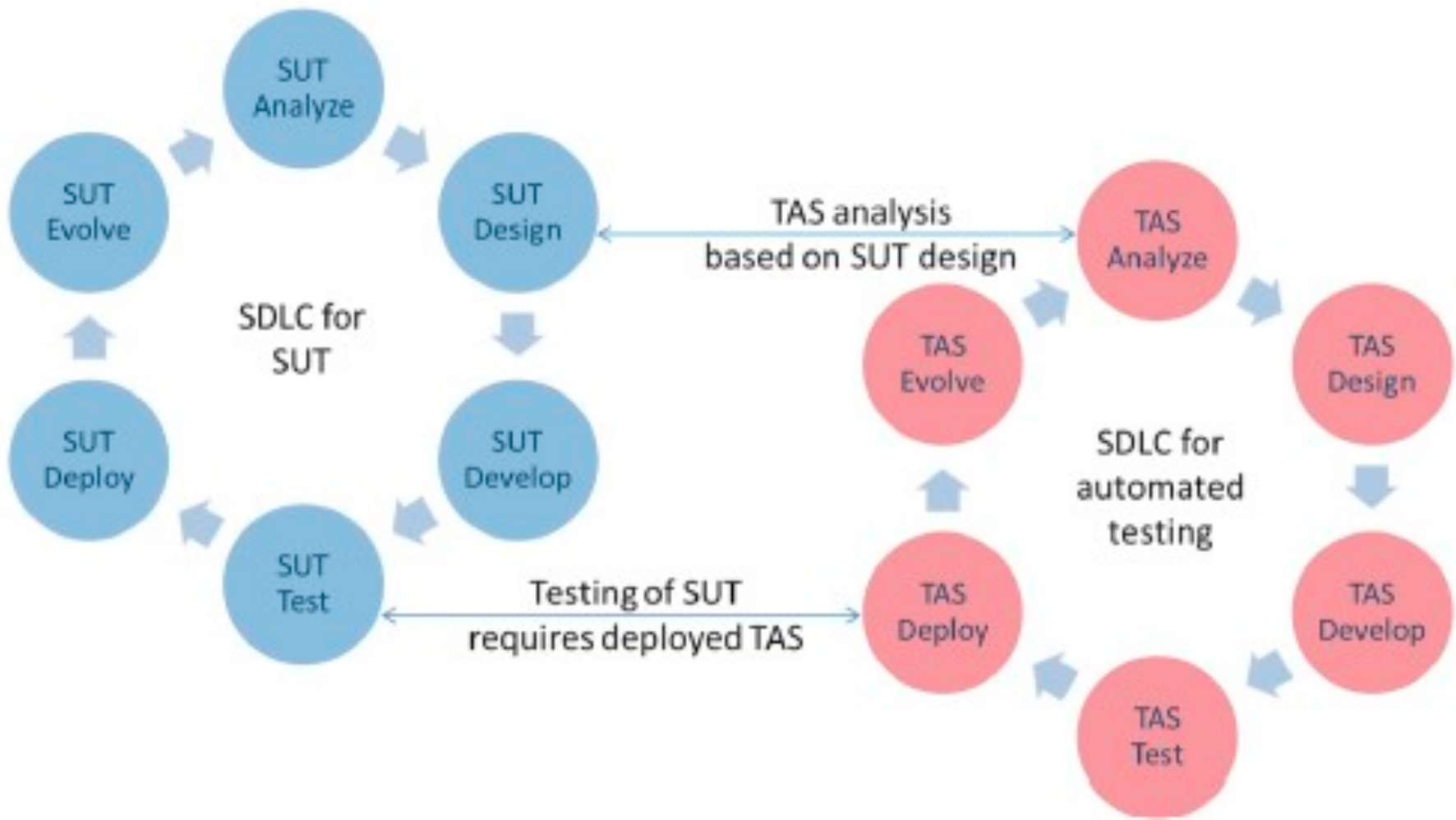
GENERIC TEST AUTOMATION ARCHITECTURE

GENERIC TEST AUTOMATION ARCHITECTURE

- Structure:
  - Test generation
  - Test definition
  - Test Execution
  - Test Adaptation

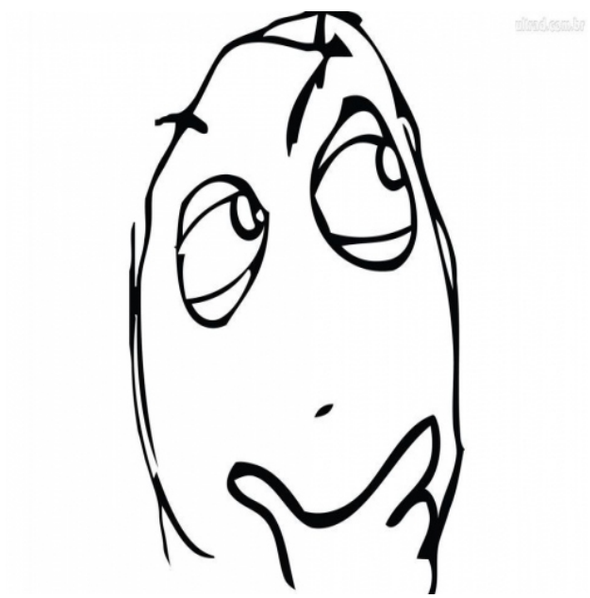
GENERIC TEST AUTOMATION ARCHITECTURE



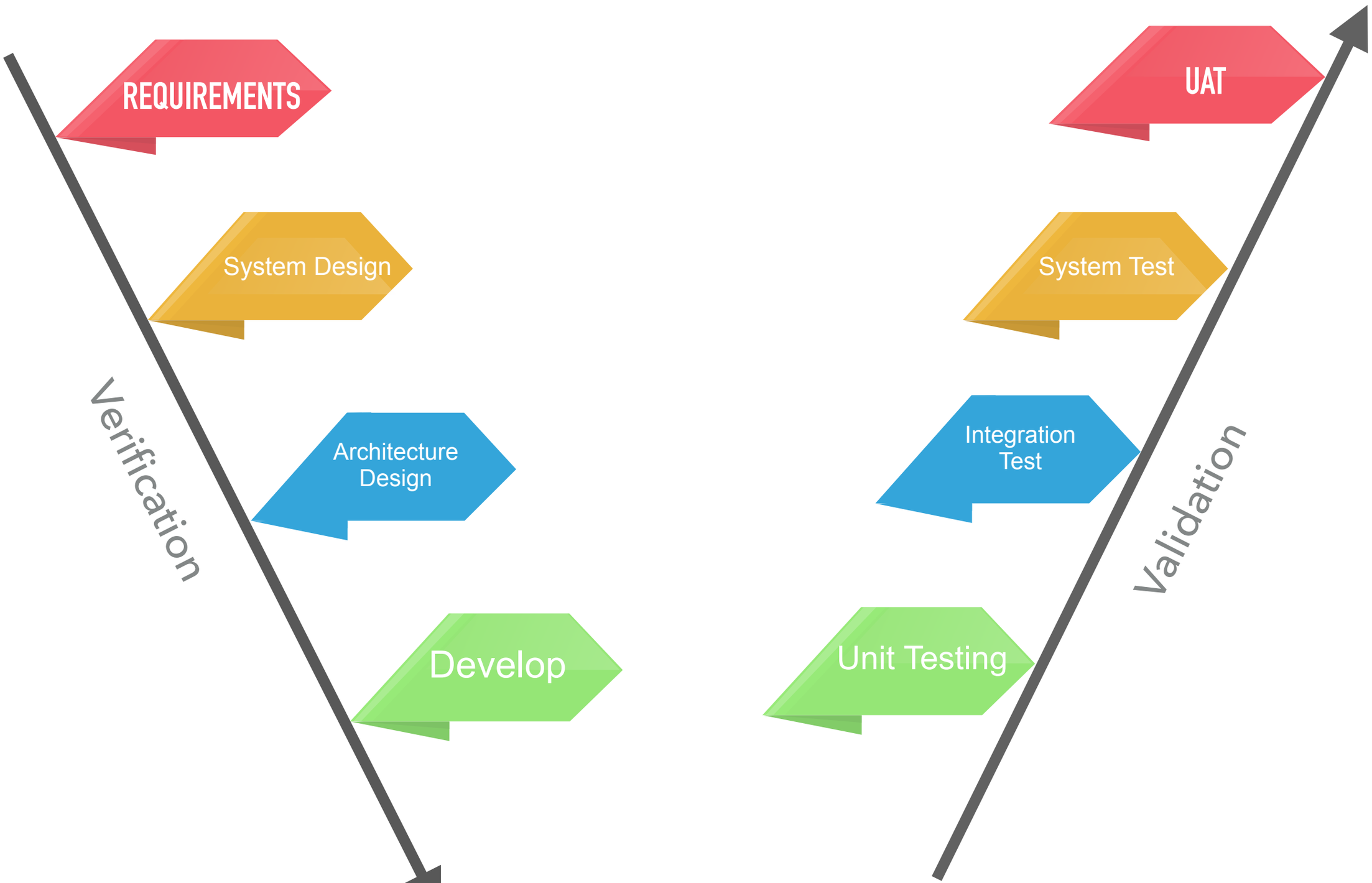

SUT Analyze
SUT Design
SUT Develop
SUT Test
SUT Deploy
SUT Evolve
SDLC for SUT
TAS analysis based on SUT design
Testing of SUT requires deployed TAS
TAS Analyze
TAS Design
TAS Develop
TAS Test
TAS Deploy
TAS Evolve
SDLC for automated testing

V MODEL

WHAT IS THE V MODEL?

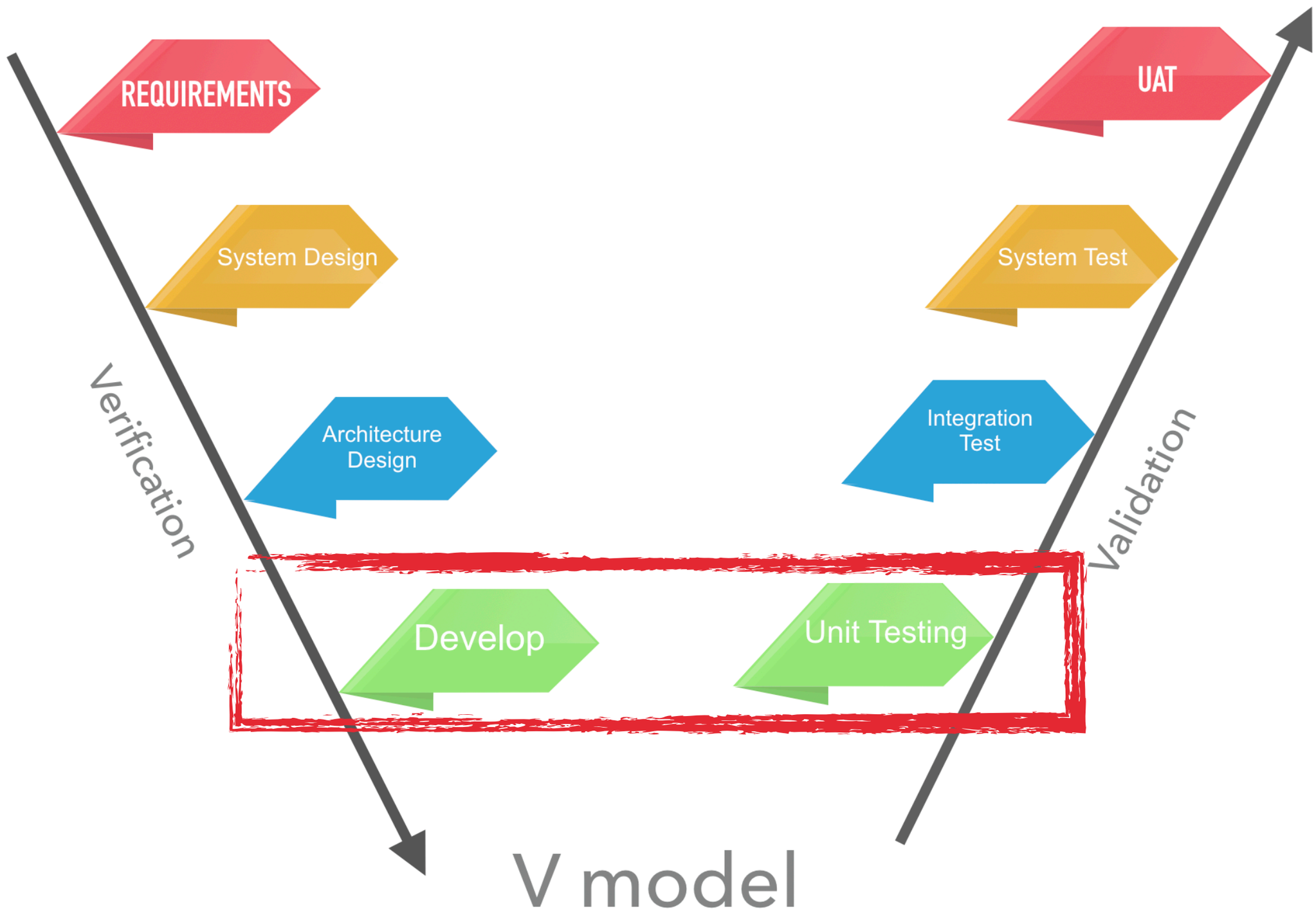


V MODEL



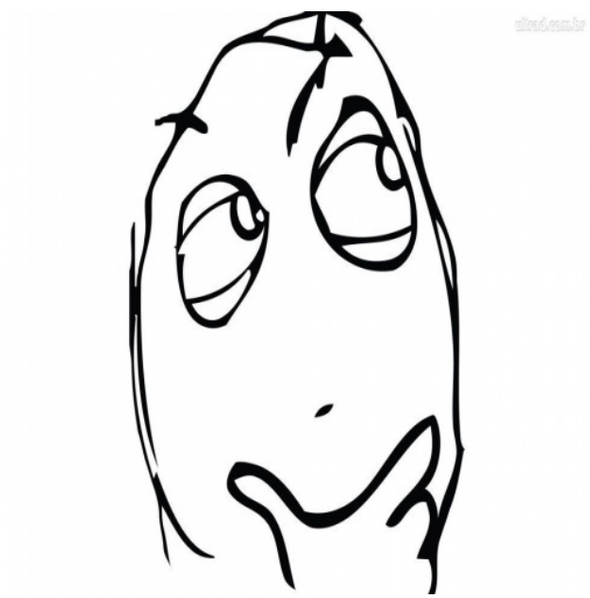
V model

V MODEL

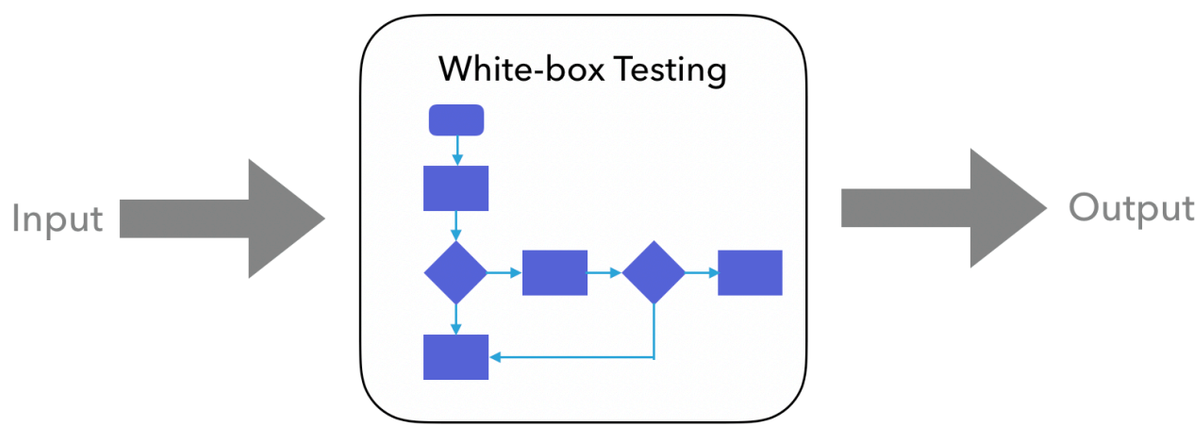

REQUIREMENTS
System Design
Architecture Design
Develop
Unit Testing
Integration Test
System Test
UAT
Verification
Validation
V model

UNIT TESTING

WHAT IS UNIT TESTING?



- First level of test pyramid.
- Testing Quadrants - Q1.
- Confidence in the component's quality.
- Find defects in the components.
- Developers.
- White-box Testing.



- Statement Testing
- Decision Testing

EXAMPLE

```java
/**
 * Created by josea.wolff on 1/19/19.
 */
public class ExampleTest {

    public String getTextInUpperCase(String text){
        return text.toUpperCase();
    }

    public String getTextInLowerCase(String text){
        return text.toLowerCase();
    }
}
```

```java
import Example.ExampleTest;
import org.testng.Assert;
import org.testng.annotations.Test;

/**
 * Created by josea.wolff on 1/19/19.
 */
public class Tests {

    @Test
    public void upperCaseTest(){
        System.out.println("\n########################## upperCaseTest ####################################");
        String expectedResult = "INTRO TO DEVOPS";
        ExampleTest ex =  new ExampleTest();
        String actualResult = ex.getTextInUpperCase("Intro to devops");
        System.out.println("Expected Result: " + expectedResult + "\nActualResult: " + actualResult);
        System.out.println("##########################################################################");
        Assert.assertEquals(actualResult,expectedResult);
    }

    @Test
    public void lowerCaseTest(){
        System.out.println("\n########################## lowerCaseTest ####################################");
        String expectedResult = "intro to devops";
        ExampleTest ex =  new ExampleTest();
        String actualResult = ex.getTextInLowerCase("Intro to devops");
        System.out.println("Expected Result: " + expectedResult + "\nActualResult: " + actualResult);
        System.out.println("##########################################################################");
        Assert.assertEquals(actualResult,expectedResult);
    }
}
```

EXAMPLE

```
Example                                                    9ms
  <no name>                                                9ms
    Tests                                                  9ms
      lowerCaseTest                                        9ms
      upperCaseTest                                        0ms
```

```
All 2 tests passed – 9ms
/Library/Java/JavaVirtualMachines/jdk1.8.0_101.jdk/Contents/Home/bin/java ...
objc[32713]: Class JavaLaunchHelper is implemented in both /Library/Java/JavaVirtualMach
  .jdk/Contents/Home/bin/java (0x106cc24c0) and /Library/Java/JavaVirtualMachines/jdk1.8
  .jdk/Contents/Home/jre/lib/libinstrument.dylib (0x106d4e4e0). One of the two will be u

########################### lowerCaseTest ####################################
Expected Result: intro to devops
ActualResult: intro to devops
##############################################################################

########################### upperCaseTest ####################################
Expected Result: INTRO TO DEVOPS
ActualResult: INTRO TO DEVOPS
##############################################################################

===============================================
Example
Total tests run: 2, Failures: 0, Skips: 0
===============================================

Process finished with exit code 0
```

EXAMPLE

```
/**
 * Created by josea.wolff on 1/19/19.
 */
public class ExampleTest {

    public String getTextInUpperCase(String text){
        return text.toUpperCase();
    }

    public String getTextInLowerCase(String text){
        return text.toUpperCase();
    }
}
```

```
Example                                  10ms
  <no name>                              10ms
    Tests                                10ms
      lowerCaseTest                      10ms
      upperCaseTest                       0ms
```

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_101.jdk/Contents/Home/bin/ja
objc[32816]: Class JavaLaunchHelper is implemented in both /Library/Jav
  and /Library/Java/JavaVirtualMachines/jdk1.8.0_101.jdk/Contents/Home/
  one is undefined.

############################ lowerCaseTest ############################
Expected Result: intro to devops
ActualResult: INTRO TO DEVOPS
###################################################################################

java.lang.AssertionError: expected [intro to devops] but found [INTRO TO DEVOPS]
Expected :intro to devops
Actual   :INTRO TO DEVOPS
 <Click to see difference>


    at org.testng.Assert.fail(Assert.java:96)
    at org.testng.Assert.failNotEquals(Assert.java:776)
    at org.testng.Assert.assertEqualsImpl(Assert.java:137)
    at org.testng.Assert.assertEquals(Assert.java:118)
    at org.testng.Assert.assertEquals(Assert.java:453)
    at org.testng.Assert.assertEquals(Assert.java:463)
    at Tests.lowerCaseTest(Tests.java:29) <16 internal calls>
    at org.testng.SuiteRunnerWorker.runSuite(SuiteRunnerWorker.java:52)
    at org.testng.SuiteRunnerWorker.run(SuiteRunnerWorker.java:84) <4 internal calls>
    at org.testng.IDEARemoteTestNG.run(IDEARemoteTestNG.java:72)
    at org.testng.RemoteTestNGStarter.main(RemoteTestNGStarter.java:127)


############################ upperCaseTest ############################
Expected Result: INTRO TO DEVOPS
ActualResult: INTRO TO DEVOPS
###################################################################################

===============================================
Example
Total tests run: 2, Failures: 1, Skips: 0
===============================================


Process finished with exit code 0
```

STATEMENT AND DECISION COVERAGE

- What are they?
- Statement Coverage
    - Executable statements in the code.
- Decision Coverage
    - Exercises the decisions in the code and tests the code that is executed based on the decision outcomes.

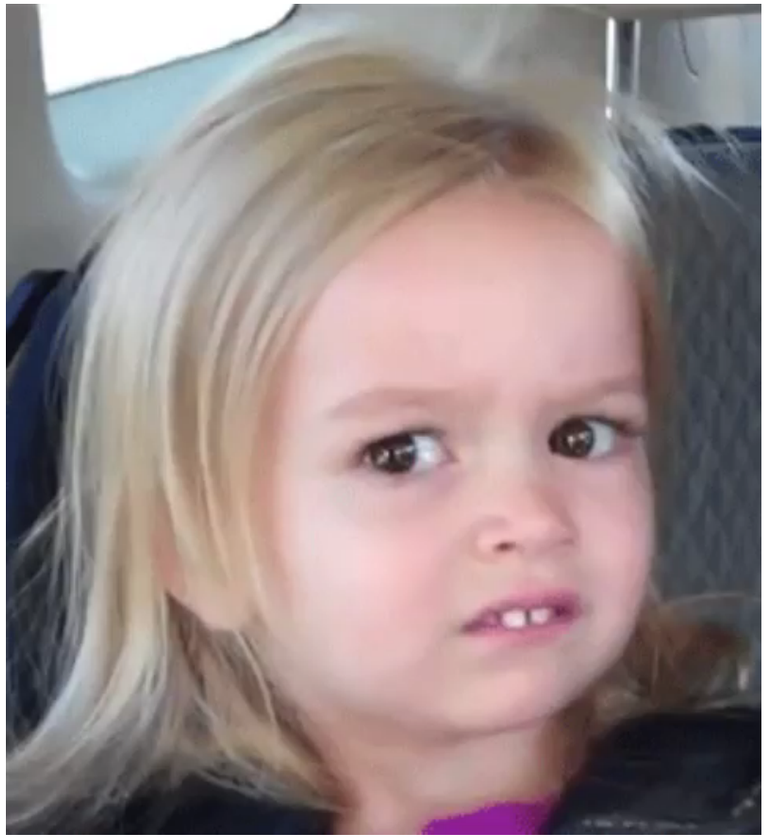
$$Statement\ Coverage = \frac{Number\ of\ executed\ statements}{Total\ number\ of\ statements} \times 100$$

$$Decision\ Coverage = \frac{Number\ of\ executed\ Decisions/Branches}{Total\ number\ of\ Decisions/Branches} \times 100$$

100% decision coverage

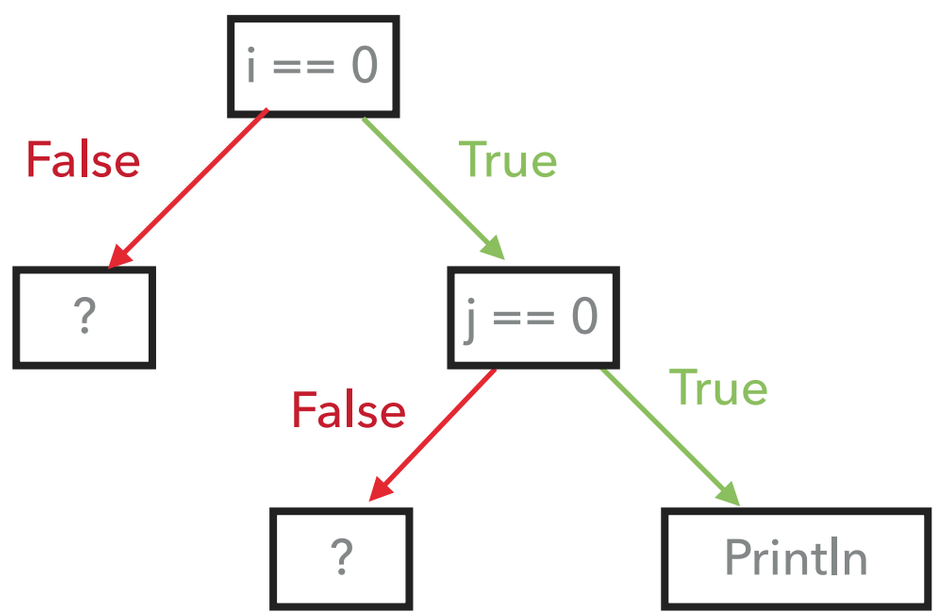
means

100% of statement coverage **not viceversa**.



EXAMPLE

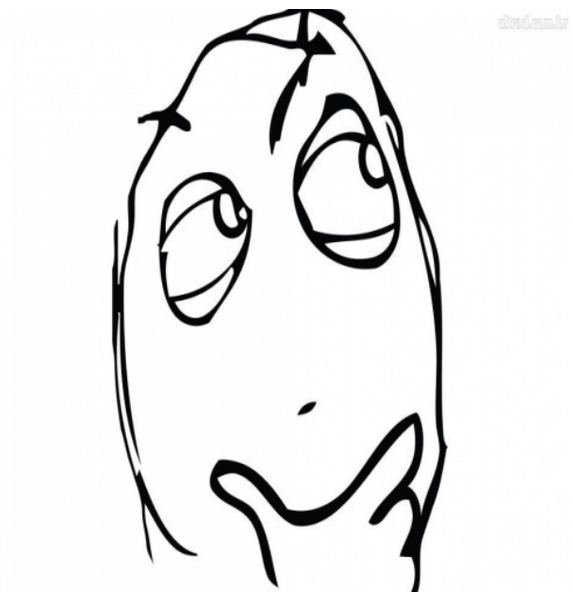
```
@Test(groups = {"Example"}, description = "Example")
public void example(){
    int i = 0;
    int j = 0;
    if(i == 0){
        if( j > 1){
            System.out.println("j is > than 1");
        }
    }
}
```



EXAMPLE

```
@Test(groups = {"Example"}, description = "Example")
public void example(){
    int i = 0;
    int j = 0;
    if(i == 0){
        if( j > 1){
            System.out.println("j is > than 1");
        }
    }
}
```

WHAT'S WRONG WITH THAT CODE?

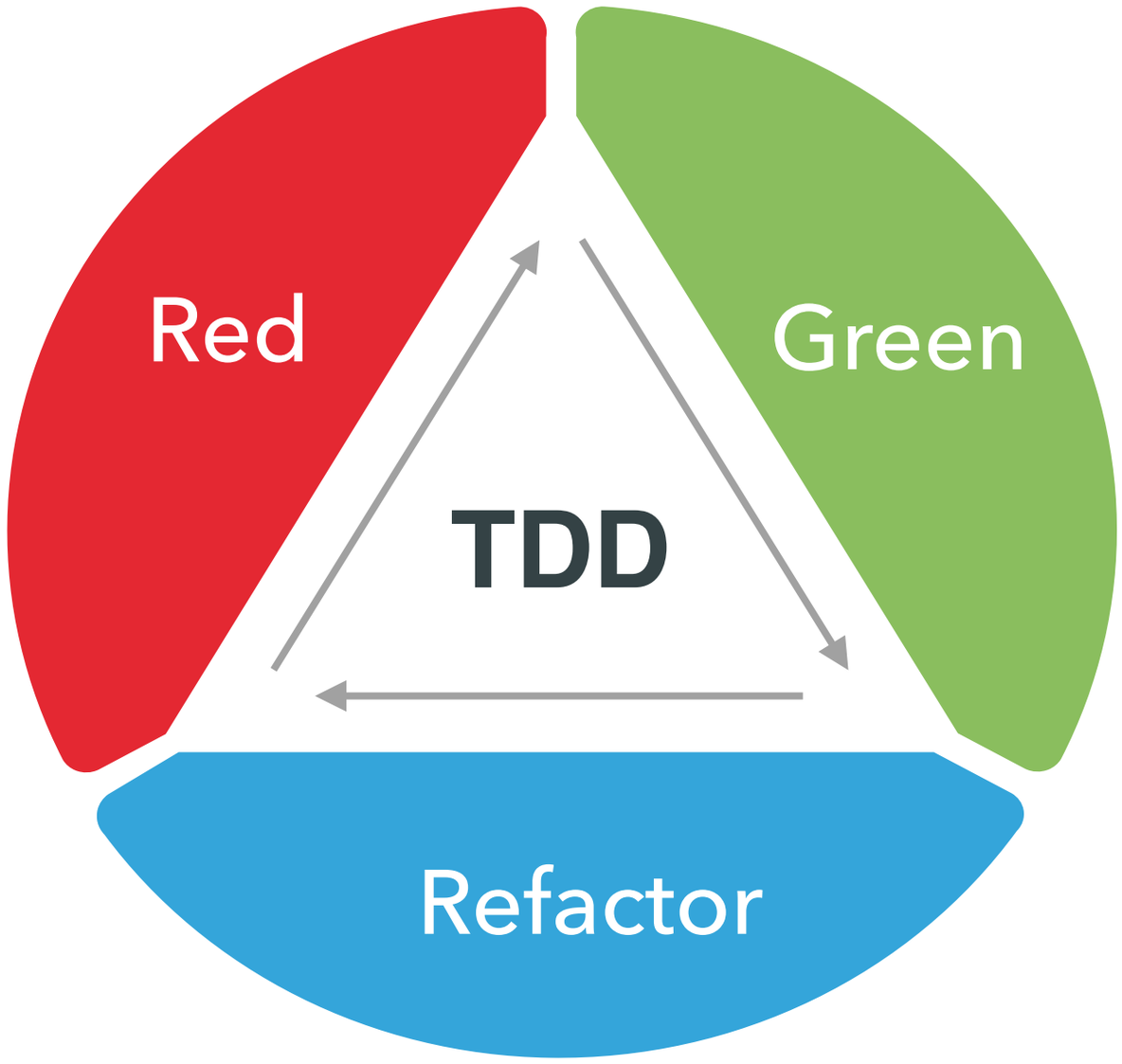


TDD

TEST DRIVEN DEVELOPMENT

- Also called Red-Green-Refactor.
- Test first programming concept.
- Pros:
    - Modular code.
    - Good architecture.
    - Documentation, etc.
- Cons:
    - Hard to write.
    - At the beginning the development could be slow.
    - Legacy code.

Test-Driven Development (TDD)



TESTNG – PARAMETERS

PARAMETERS

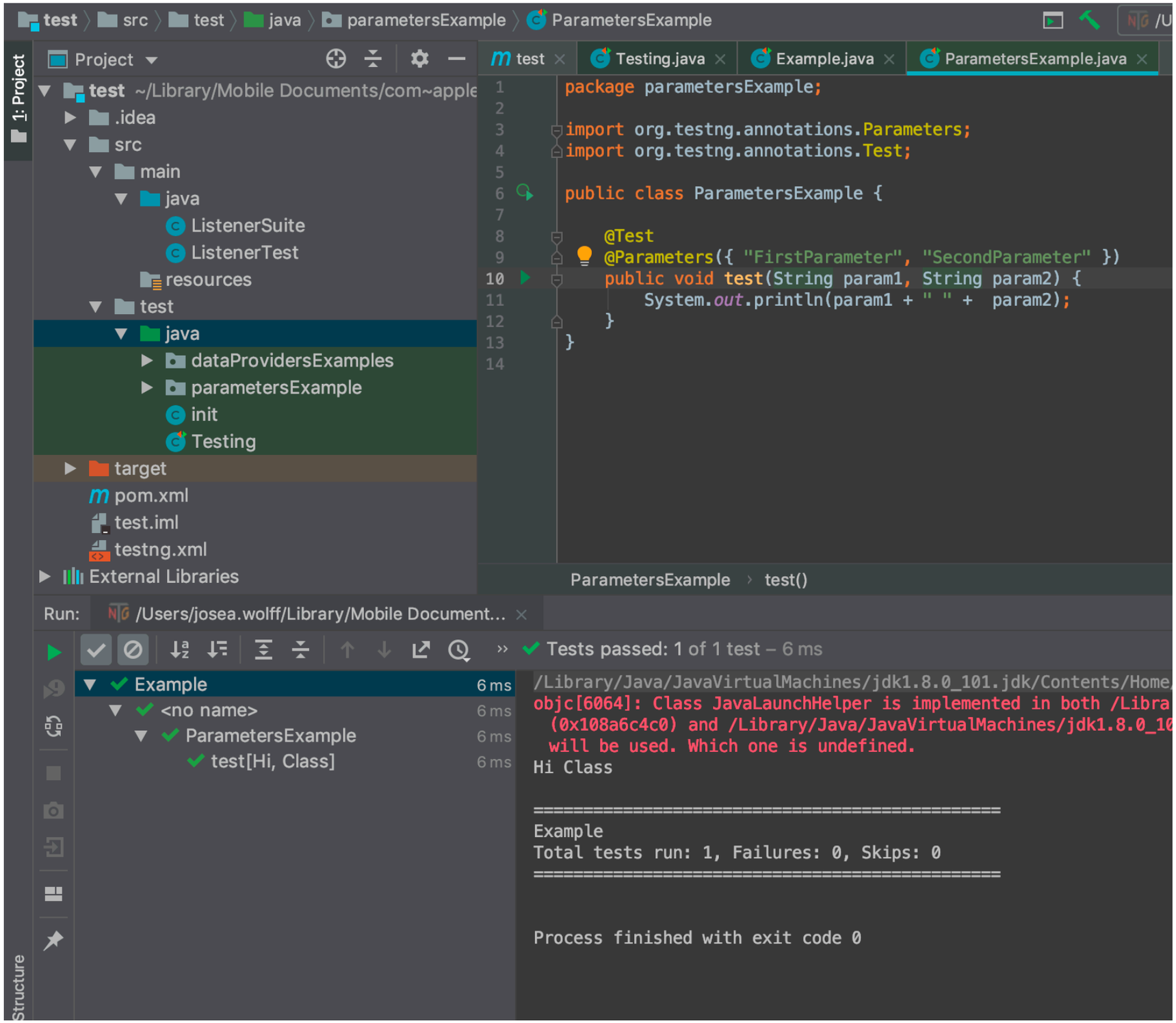
- Run multiple times the same test with different parameters.
- Specify in the testng.xml file.
- @Before/ @After and @Factory annotations.
- Can be optionals (@Optional).

PARAMETERS – EXAMPLE

```java
package parametersExample;

import org.testng.annotations.Parameters;
import org.testng.annotations.Test;

public class ParametersExample {

    @Test
    @Parameters({ "FirstParameter", "SecondParameter" })
    public void test(String param1, String param2) {
        System.out.println(param1 + " " +  param2);
    }
}
```

```xml
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Example" preserve-order="true" allow-return-values="true" verbose="1">
    <parameter name="FirstParameter" value="Hi"/>
    <parameter name="SecondParameter" value="Class"/>
    <test name="Example">
        <classes>
            <class name="parametersExample.ParametersExample"/>
        </classes>
    </test>

    <listeners>
        <listener class-name="ListenerTest"/>
        <listener class-name="ListenerSuite"/>
    </listeners>
</suite>
```

PARAMETERS – EXAMPLE



PARAMETERS – EXAMPLE2 (@OPTIONAL)

```java
package parametersExample;

import org.testng.annotations.Optional;
import org.testng.annotations.Parameters;
import org.testng.annotations.Test;

public class ParametersExample {

    @Test
    @Parameters({ "FirstParameter", "SecondParameter" })
    public void test(@Optional String param1, String param2) {
        System.out.println(param1 + " " +  param2);
    }
}
```

```xml
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Example" preserve-order="true" allow-return-values="true" verbose="1">
    <!--<parameter name="FirstParameter" value="Hi"/>-->
    <parameter name="SecondParameter" value="Class"/>
    <test name="Example">
        <classes>
            <class name="parametersExample.ParametersExample"/>
        </classes>
    </test>

    <listeners>
        <listener class-name="ListenerTest"/>
        <listener class-name="ListenerSuite"/>
    </listeners>
</suite>
```

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_101.jdk/Co
objc[6105]: Class JavaLaunchHelper is implemented in
  (0x10d56e4c0) and /Library/Java/JavaVirtualMachines/
  will be used. Which one is undefined.
null Class

===============================================
Example
Total tests run: 1, Failures: 0, Skips: 0
===============================================


Process finished with exit code 0
```

PARAMETERS – EXAMPLE2 (@OPTIONAL)

```java
public class ParametersExample {

    @Test
    @Parameters({ "FirstParameter", "SecondParameter" })
    public void test(@Optional("HI") String param1, String param2) {
        System.out.println(param1 + " " +  param2);
    }
}
```

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_101.j
objc[6135]: Class JavaLaunchHelper is implemente
  (0x10f21c4c0) and /Library/Java/JavaVirtualMach
  will be used. Which one is undefined.
HI Class

===============================================
Example
Total tests run: 1, Failures: 0, Skips: 0
===============================================


Process finished with exit code 0
```

```xml
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Example" preserve-order="true" allow-return-values="true" verbose="1">
    <!--<parameter name="FirstParameter" value="Hi"/>-->
    <parameter name="SecondParameter" value="Class"/>
    <test name="Example">
        <classes>
            <class name="parametersExample.ParametersExample"/>
        </classes>
    </test>

    <listeners>
        <listener class-name="ListenerTest"/>
        <listener class-name="ListenerSuite"/>
    </listeners>
</suite>
```

TESTNG – DATAPROVIDERS

DATAPROVIDERS

- What inconvenient can you see with the parameters?
- testng.xml is not sufficient.
- Complex parameters (Example: API).
- Method that returns an array of array of objects.
- @DataProvider.
- Specify in the @Test.

DATAPROVIDER – EXAMPLE (STATIC STRUCTURE)

test 〉 src 〉 test 〉 java 〉 dataProvidersExamples 〉 Example

Project

- test ~/Library/Mobile Documents/com~apple
  - .idea
  - src
    - main
      - java
        - ListenerSuite
        - ListenerTest
      - resources
    - test
      - java
        - dataProvidersExamples
        - parametersExample
        - init
        - Testing
  - target
  - pom.xml
  - test.iml
  - testng.xml
- External Libraries

test | Testing.java | Example.java | testng.xml

```java
package dataProvidersExamples;

import org.testng.annotations.DataProvider;
import org.testng.annotations.Test;

public class Example {

    @DataProvider(name = "ExampleDP")
    public static Object[][] ExampleDP() {
        return new Object[][] { { "1", "1" }, { "2", "2" }};
    }

    @Test(dataProvider = "ExampleDP")
    public void exampleDP(String firstParameter, String secondParameters){
        System.out.println(firstParameter + " " + secondParameters);
    }
}
```

Run: Example.exampleDP

Tests passed: 2 of 2 tests – 6 ms

Default Suite	6 ms
test	6 ms
Example	6 ms
exampleDP[1, 1]	6 ms
exampleDP[2, 2] (1)	0 ms

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_101.jdk/Contents/Home/bin/java ...
objc[6157]: Class JavaLaunchHelper is implemented in both /Library/Java/JavaVirtu
  (0x1023144c0) and /Library/Java/JavaVirtualMachines/jdk1.8.0_101.jdk/Contents/H
  will be used. Which one is undefined.
1 1
2 2

===============================================
Default Suite
Total tests run: 2, Failures: 0, Skips: 0
===============================================


Process finished with exit code 0
```

DATAPROVIDER – EXAMPLE (DYNAMIC STRUCTURE)

test 〉 src 〉 test 〉 java 〉 dataProvidersExamples 〉 ExampleDynamicStructure

Project

- test ~/Library/Mobile Documents/com~apple
  - .idea
  - src
    - main
      - java
        - ListenerSuite
        - ListenerTest
      - resources
    - test
      - java
        - dataProvidersExamples
          - ExampleDynamicStructure
          - ExampleStaticStructure
        - parametersExample
        - init
        - Testing
  - target
  - pom.xml
  - test.iml

test | Testing.java | ExampleStaticStructure.java | ExampleDynamicStructure.java

```
1   package dataProvidersExamples;
2
3   import ...
10
11  public class ExampleDynamicStructure {
12
13      @DataProvider(name = "ExampleDP")
14      public Iterator<Object[]> locales() {
15          List<Object[]> data = new ArrayList<Object[]>();
16          for (int i = 0; i < 15; i++){
17              data.add(new Integer[]{i});
18          }
19          return data.iterator();
20      }
21
22      @Test(dataProvider = "ExampleDP")
23      public void exampleDP(int number){
24          if(number <= 10){
25              System.out.println(number );
26          }else{
27              Assert.fail( s: number + " is greater than 10");
28          }
29
```

ExampleDynamicStructure 〉 exampleDP()

Run: ExampleDynamicStructure.exampleDP

Tests failed: 4, passed: 11 of 15 tests – 9 ms

Default Suite	9 ms
test	9 ms
ExampleDynamicStructure	9 ms
exampleDP[0]	4 ms
exampleDP[1] (1)	0 ms
exampleDP[2] (2)	0 ms
exampleDP[3] (3)	1 ms
exampleDP[4] (4)	0 ms
exampleDP[5] (5)	1 ms
exampleDP[6] (6)	0 ms
exampleDP[7] (7)	0 ms
exampleDP[8] (8)	0 ms
exampleDP[9] (9)	0 ms
exampleDP[10] (10)	1 ms
exampleDP[11] (11)	2 ms
exampleDP[12] (12)	0 ms
exampleDP[13] (13)	0 ms
exampleDP[14] (14)	0 ms

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_101.jdk/Contents/Home/bin/java ...
objc[6189]: Class JavaLaunchHelper is implemented in both /Library/Java/JavaVir
  (0x10bd194c0) and /Library/Java/JavaVirtualMachines/jdk1.8.0_101.jdk/Contents
  will be used. Which one is undefined.
0
1
2
3
4
5
6
7
8
9
10

java.lang.AssertionError: 11 is greater than 10

    at org.testng.Assert.fail(Assert.java:96)
    at dataProvidersExamples.ExampleDynamicStructure.exampleDP(ExampleDynamicSt
    at org.testng.SuiteRunnerWorker.runSuite(SuiteRunnerWorker.java:52)
    at org.testng.SuiteRunnerWorker.run(SuiteRunnerWorker.java:84) <4 internal
    at org.testng.IDEARemoteTestNG.run(IDEARemoteTestNG.java:73)
    at org.testng.RemoteTestNGStarter.main(RemoteTestNGStarter.java:123)
```

EXAMPLE

- Calculator
  - +
  - -
  - *
  - /

LET'S
CODE
.IO

THANK YOU.