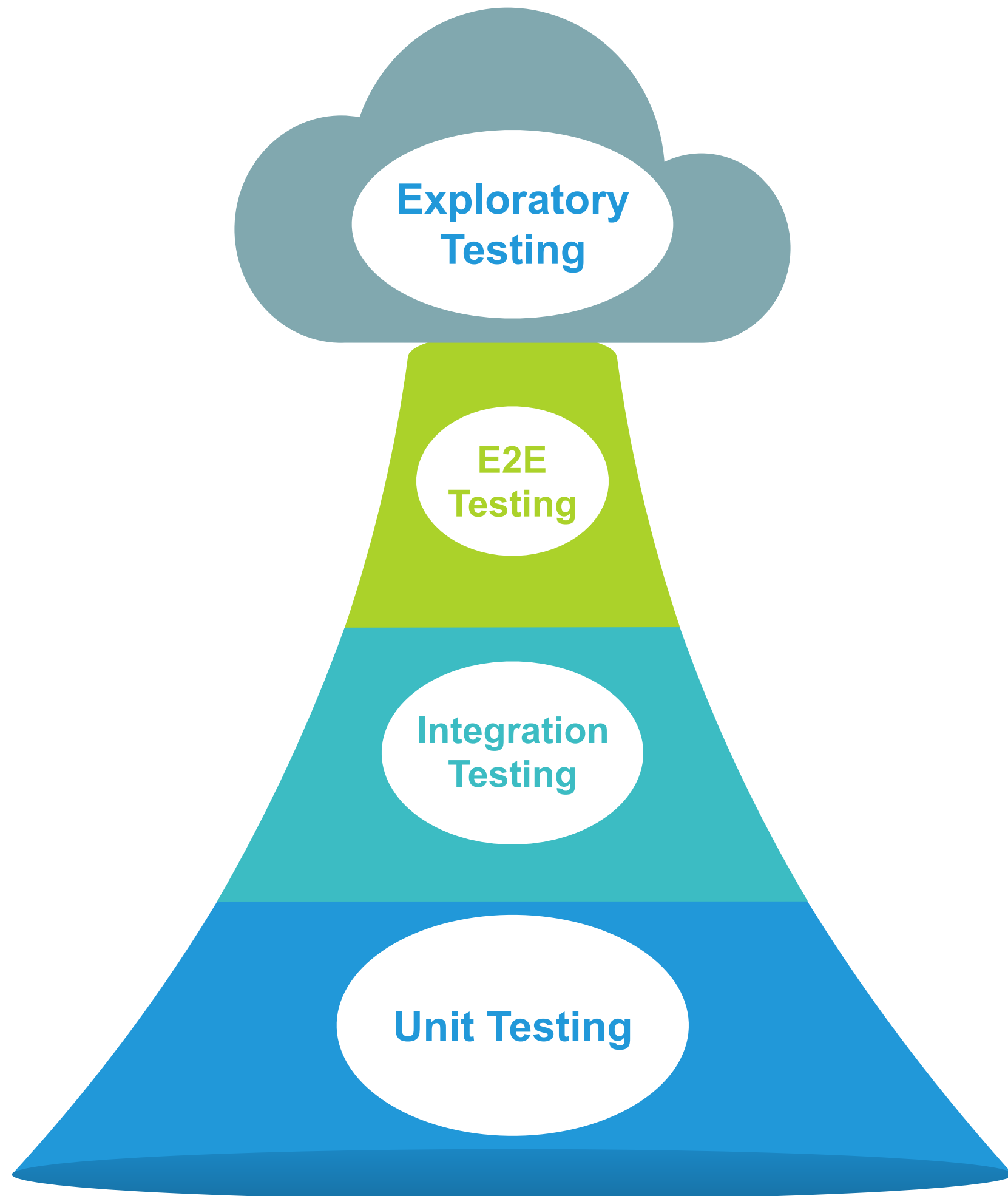

TESTING GROUPS USING TESTING TOOLS





- ▶ Build automation tool.
- ▶ Used primary for Java.
- ▶ Two aspects:
 - ▶ How the software is built.
 - ▶ Dependencies.
- ▶ XML file.

- ▶ Groupid - Project identifier across all projects.
- ▶ Artifactid - Jar name without version.
- ▶ Version - Jar version (if you distribute it)

POM File

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.ufm.introdevops</groupId>
  <artifactId>testing</artifactId>
  <version>1.0</version>

  <dependencies>
    <!-- ***** AW DEPENDENCIES *****-->
    <!-- https://mvnrepository.com/artifact/org.testng/testng -->
    <dependency>
      <groupId>org.testng</groupId>
      <artifactId>testng</artifactId>
      <version>6.14.3</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

TestNG

- ▶ Open source automation testing framework.
- ▶ Inspired from JUnit and NUnit but introducing new functionalities.
- ▶ Key Features:
 - ▶ Groups
 - ▶ Reports (Basic)
 - ▶ Integrated with other tools (Jenkins, Maven)
 - ▶ Annotations.

TestNG

- ▶ Testng.xml
 - ▶ Configuration file in testng.
 - ▶ Define test suites.
 - ▶ Include or exclude.
 - ▶ Parallelism.

```
<?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">
  <test name="Example">
    <groups>
      <run>
        <exclude name="brokenTests" />
        <include name="regressionTests" />
      </run>
    </groups>
    <classes>
      <class name="Tests"/>
    </classes>
  </test>
</suite>
```

TestNG

► Common annotations:

@BeforeSuite: The annotated method will be run before all tests in this suite have run.

@BeforeTest:	The annotated method will be run before any test method belonging to the classes inside the tag is run.
---------------------	---

@BeforeClass: The annotated method will be run before the first test method in the current class is invoked.

@BeforeMethod:	The annotated method will be run before each test method.
-----------------------	---

@Test: The test method or class

@AfterMethod:	The annotated method will be run after each test method.
----------------------	--

@AfterClass: The annotated method will be run after all the test methods in the current class have been run.

@AfterTest:	The annotated method will be run after all the test methods belonging to the classes inside the tag have run.
--------------------	---

@AfterSuite: The annotated method will be run after all tests in this suite have run.

TestNG

▶ @Test Annotations attributes:

- ▶ **description:** A string that gives information about the test it is attached to.
- ▶ **TimeOut:** This is the maximum number of milliseconds for a test run.
- ▶ **priority:** When this attribute is not declared in your tests, the order in which the test methods are run is random.
- ▶ **dependsOnMethods:** When you want to run a test only after another test has run successfully.
- ▶ **enabled:** This attribute has boolean values. This attribute is only worth specifying explicitly when you do not want a certain test method to be run, by setting the attribute to false
- ▶ **groups:** This attribute is useful for grouping together tests that relate to the same functionality, are of the same importance or are of the same type.

TestNG

```
@Test(description = "Upper Case testt",
      timeout = 20000,
      priority = 1,
      dependsOnMethods = "lowerCaseTest",
      groups = {"example", "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);
}
```

More annotations that we will cover in the next classes...

HOW CAN I DECIDE IF THE TEST PASS OR FAIL?



TestNG

- ▶ Assertions:
 - ▶ Hard Assert.
 - ▶ Soft Assert.

WHAT COULD BE THE DIFFERENCE?



TestNG

- ▶ Hard Assert:
 - ▶ Throw an `AssertException` after the test fails and the test is marked as failed.
 - ▶ Next test is executed.

TESTING GROUPS USING TESTING TOOLS

```
import ...

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

    @Test(description = "Upper Case testt",
            timeout = 20000,
            priority = 1,
            //dependsOnMethods = "lowerCaseTest",
            groups = {"example", "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);
        System.out.println("--> Hard assertion");
    }
}
```

Tests > upperCaseTest()

Tests passed: 1 of 1 test – 17 ms

/Library/Java/JavaVirtualMachines/jdk1.8.0_101.jdk/Contents/Home/bin/java ...

objc[1087]: Class JavaLaunchHelper is implemented in both /Library/Java/JavaVirtualMachines/jdk1.8.0_101.jdk/Contents/Home/bin/java (0x10f76d4c0) and /Library/Java/JavaVirtualMachines/jdk1.8.0_101.jdk/Contents/Home/jre/lib/libinstrument.dylib (0x10f7e74e0). One of the two will be used. Which one is undefined.

upperCaseTest

Expected Result: INTRO TO DEVOPS

ActualResult: INTRO TO DEVOPS

#####

--> Hard assertion

=====
Example

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

=====

Process finished with exit code 0

TESTING GROUPS USING TESTING TOOLS

```
8  public class Tests {
9
10     @Test(description = "Upper Case testt",
11            timeout = 20000,
12            priority = 1,
13            //dependsOnMethods = "lowerCaseTest",
14            groups = {"example", "test"})
15     public void upperCaseTest(){
16         System.out.println("\n##### upperCaseTest #####");
17         String expectedResult = "INTRr0 TO DEVOPS";
18         ExampleTest ex = new ExampleTest();
19         String actualResult = ex.getTextInUpperCase("Intro to devops");
20         System.out.println("Expected Result: " + expectedResult + "\nActualResult: " + actualResult);
21         System.out.println("#####");
22         Assert.assertEquals(actualResult, expectedResult);
23         System.out.println("--> Hard assertion");
24     }
25 }
```

Tests > upperCaseTest()

ment... x

>> Tests failed: 1 of 1 test – 16 ms

```
16 ms ##### upperCaseTest #####
16 ms Expected Result: INTRr0 TO DEVOPS
16 ms ActualResult: INTRO TO DEVOPS
16 ms #####
java.lang.AssertionError: expected [INTRr0 TO DEVOPS] but found [INTRO TO DEVOPS]
Expected :INTRr0 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.upperCaseTest(Tests.java:22) <7 internal calls>
at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
at java.lang.Thread.run(Thread.java:745)
```

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

?????

TestNG

- ▶ Soft Assert:
 - ▶ Collect errors during @Test.
 - ▶ Does not throw an `AssertionException`.
 - ▶ Continue with the next step after an assertion happens.

TESTING GROUPS USING TESTING TOOLS

Project ▾

Java Projects [Examples] ~/Library/Mok

▸ .idea

▾ src

▾ main

▾ java

▸ Example

resources

▾ test

▾ java

Tests

target

test-output

Examples.iml

m pom.xml

testng.xml

External Libraries

Scratches and Consoles

testing x

testng.xml x

Tests.java x

ExampleTest.java x

8 */

9 public class Tests {

10

11 @Test(description = "Upper Case testt",

12 timeout = 20000,

13 priority = 1,

14 //dependsOnMethods = "lowerCaseTest",

15 groups = {"example", "test"})

16 public void upperCaseTest(){

17 SoftAssert softAssert = new SoftAssert();

18 System.out.println("\n##### upperCaseTest #####");

19 String expectedResult = "INTRO TO DEVOPS";

20 ExampleTest ex = new ExampleTest();

21 String actualResult = ex.getTextInUpperCase("Intro to devops");

22 System.out.println("Expected Result: " + expectedResult + "\nActualResult: " + actualResult);

23 System.out.println("#####");

24 softAssert.assertEquals(actualResult,expectedResult);

25 softAssert.assertEquals(s: "Fail test",expectedResult);

26 System.out.println("--> Continue");

27 System.out.println("--> Hard assertion");

28 }

29

30 @Test

31 public void lowerCaseTest(){

32 System.out.println("\n##### lowerCaseTest #####");

Tests > upperCaseTest()

Run: /Users/josea.wolf/Library/Mobile Document...

Tests passed: 1 of 1 test – 20 ms

Example 20 ms

<no name> 20 ms

Tests 20 ms

upperCaseTest 20 ms

/Library/Java/JavaVirtualMachines/jdk1.8.0_101.jdk/Contents/Home/bin/java ...

objc[1122]: Class JavaLaunchHelper is implemented in both /Library/Java/JavaVirtualMachines/jdk1.8.0_101.jdk/Contents/Hon

(0x1050cb4c0) and /Library/Java/JavaVirtualMachines/jdk1.8.0_101.jdk/Contents/Home/jre/lib/libinstrument.dylib (0x10514

will be used. Which one is undefined.

upperCaseTest

Expected Result: INTRO TO DEVOPS

ActualResult: INTRO TO DEVOPS

#####

--> Continue

--> Hard assertion

=====

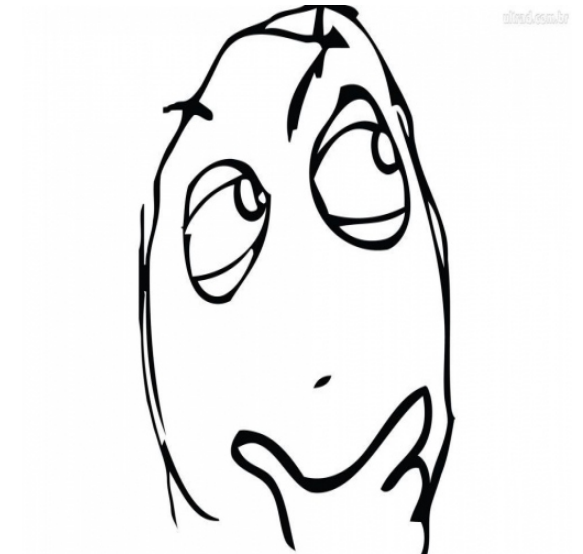
Example

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

=====

?????

WHY THE TEST NOT FAILED?



- ▶ `Assert.assertAll()` - Last Statement in the `@Test`.

TESTING GROUPS USING TESTING TOOLS

The screenshot displays an IDE interface for a Java project named 'Java Projects [Examples]'. The project structure on the left shows a 'test' directory containing a 'java' subdirectory with a 'Tests' class. The main editor shows the 'Tests.java' file with the following code:

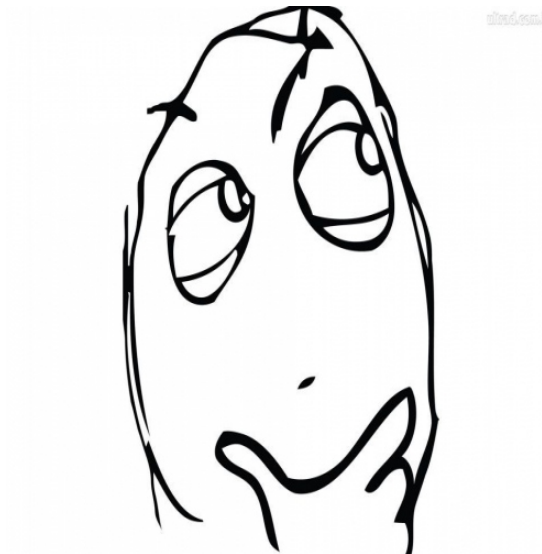
```
1 import ...
5
6 /**
7  * Created by josea.wolff on 1/19/19.
8  */
9 public class Tests {
10
11     @Test(description = "Upper Case testt",
12            timeout = 20000,
13            priority = 1,
14            //dependsOnMethods = "lowerCaseTest",
15            groups = {"example", "test"})
16     public void upperCaseTest(){
17         SoftAssert softAssert = new SoftAssert();
18         System.out.println("\n##### upperCaseTest #####");
19         String expectedResult = "INTRO TO DEVOPS";
20         ExampleTest ex = new ExampleTest();
21         String actualResult = ex.getTextInUpperCase("Intro to devops");
22         System.out.println("Expected Result: " + expectedResult + "\nActualResult: " + actualResult);
23         System.out.println("#####");
24         softAssert.assertEquals(actualResult, expectedResult);
25         softAssert.assertEquals(s: "Fail test", expectedResult);
26         System.out.println("--> Continue");
27         System.out.println("--> Hard assertion");
28         softAssert.assertAll();
29     }
```

A red arrow points to the `softAssert.assertAll();` line (line 28). The run console at the bottom shows the test results:

```
Run: /Users/josea.wolff/Library/Mobile Document... x
Tests failed: 1 of 1 test - 16 ms
Example 16 ms
  <no name> 16 ms
    Tests 16 ms
      upperCaseTest 16 ms
        objc[1143]: Class JavaLaunchHelper is implemented in both /Library/Java/JavaVirtualMachines/jdk1.8.0_101.jdk/Contents
        (0x10534d4c0) and /Library/Java/JavaVirtualMachines/jdk1.8.0_101.jdk/Contents/Home/jre/lib/libinstrument.dylib (0x1
        will be used. Which one is undefined.
        ##### upperCaseTest #####
        Expected Result: INTRO TO DEVOPS
        ActualResult: INTRO TO DEVOPS
        #####
        --> Continue
        --> Hard assertion
        java.lang.AssertionError: The following asserts failed:
          expected [INTRO TO DEVOPS] but found [Fail test]
        Expected :INTRO TO DEVOPS
        Actual   :Fail test
        <Click to see difference>
        at org.testng.asserts.SoftAssert.assertAll(SoftAssert.java:43)
        at Tests.upperCaseTest(Tests.java:28) <7 internal calls>
        at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
        at java.util.concurrent.FutureTask.run(FutureTask.java:266)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
```

A large green checkmark is overlaid on the right side of the console output.

WHEN YOU SHOULD USE A SOFT ASSERT AND HARD ASSERT?







THANK YOU.