Test Frameworks: JUnit & TestNG

Kedar Mhaswade

07 Mar 2006

Note: Each presentation should contain a title slide (either with or without photo) and an ending slide. Titles on these pages should be in ALL CAP format.

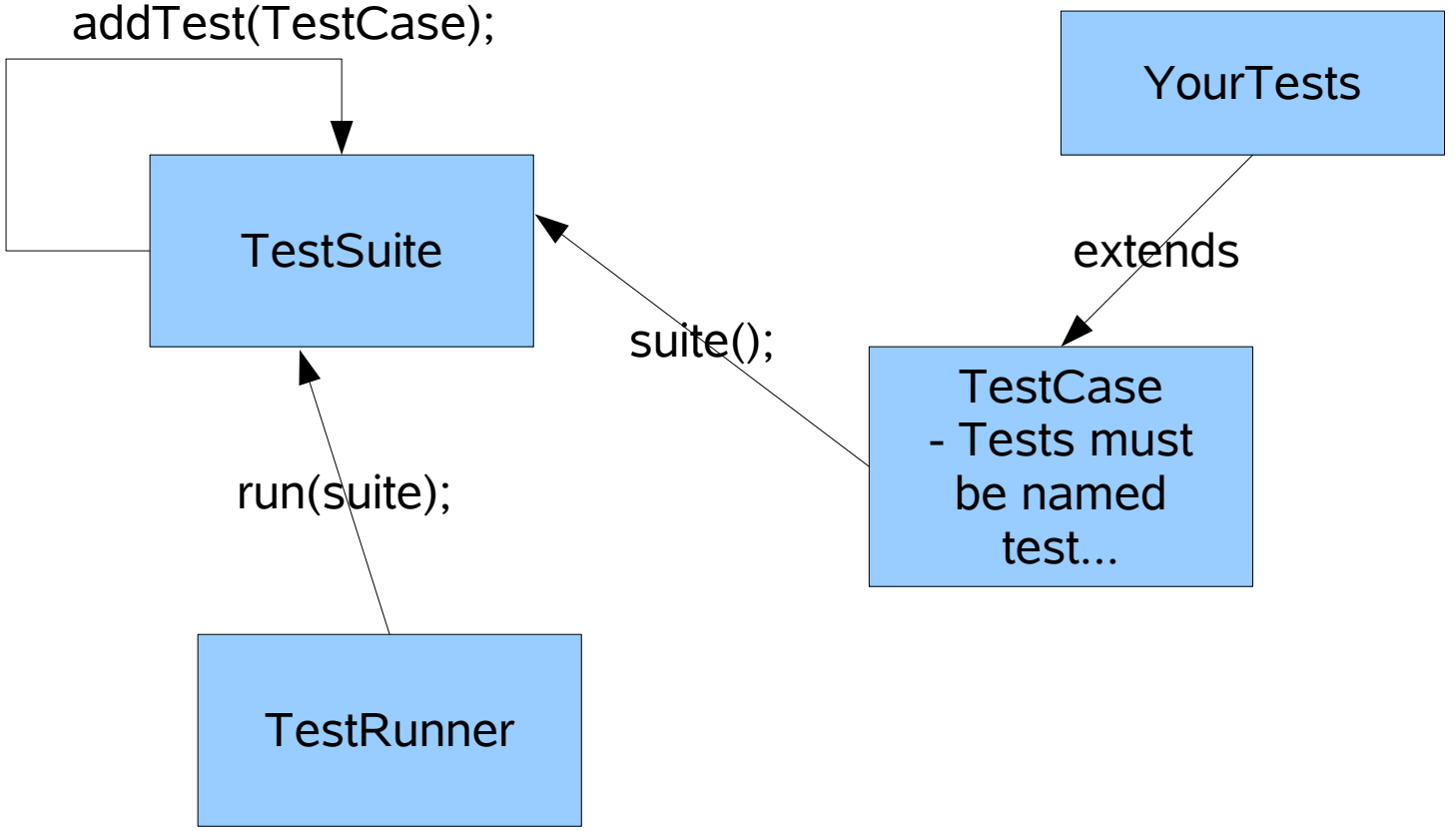
AGENDA

- JUnit – Philosophy
- JUnit-4 Changes from JUnit-3
- JUnit - Features
- TestNG – Philosophy
- TestNG - Features
- Comparison of JUnit-4 and TestNG
- Recommendations
- Beyond JUnit and TestNG

JUnit Philosophy

- Human Judgment Should not be Needed to “Test”
- Two Basic Classes: TestCase and TestSuite
- Write a Simple Test Class for Each Class to be Tested
  - > Not Necessary, but More Manageable
  - > YourTestCase Extends JUnit TestCase
- TestSuite comprises of TestCases and other TestSuites
- Minimal “Framework Code”

JUnit: Important Classes



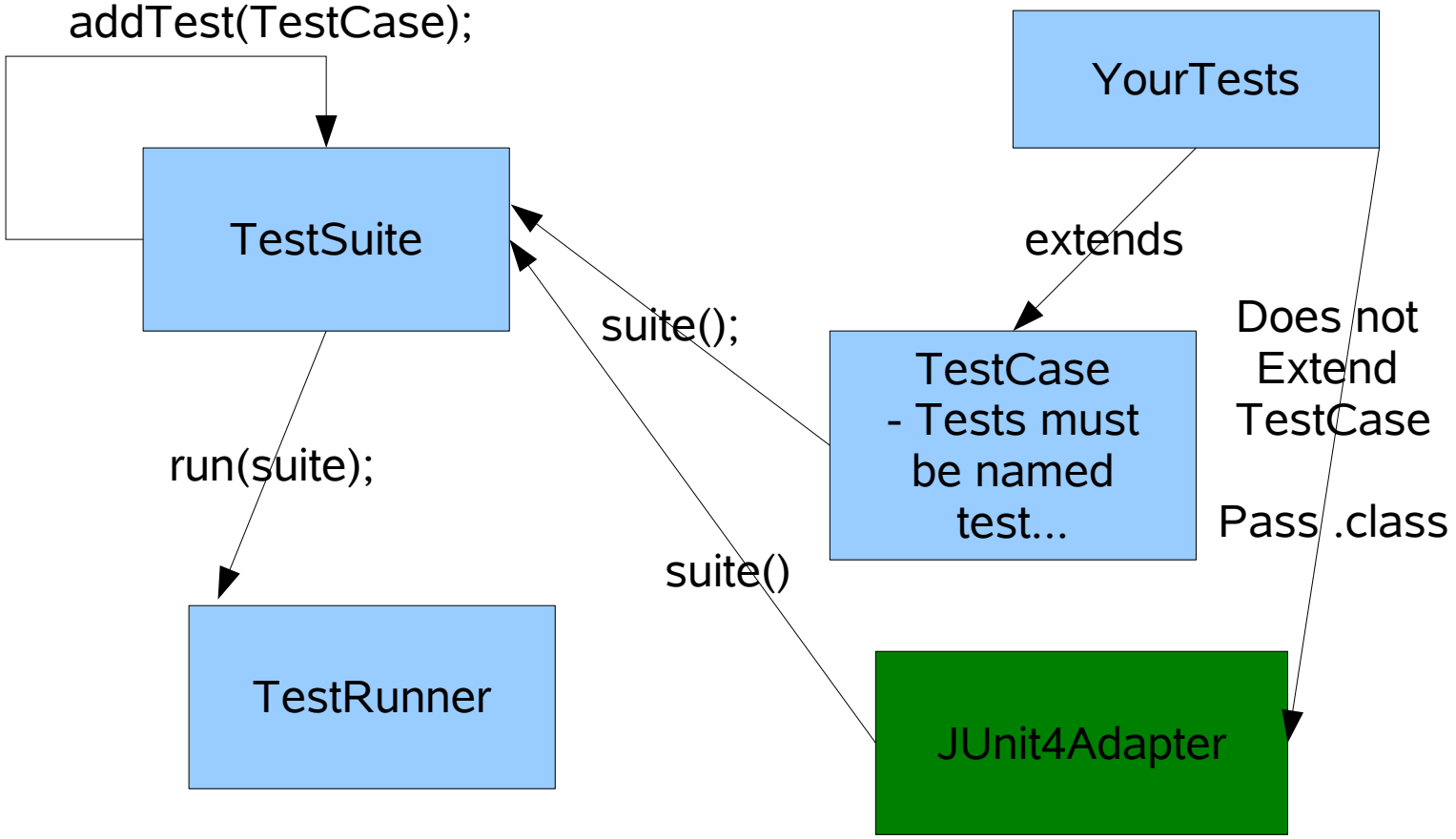
JUnit4: Changes

- Introduces Few Annotations
- Your Test does not Have to Extend j.f.TestCase
- An Adapter is Introduced for Running with Old Runners
  - > junit.framework.JUnit4TestAdapter
- No Change that **has** to be Adopted by Users
  - > Backward Compatibility is of Essense
- j.f.ActiveTestSuite
  - > Runs Tests in Separate Thread
- A Test can be Timed Out or Expected to Throw an Exception

JUnit4: More Changes

- Testing Exceptions is Easier (@Test(expected=))
- Timed Tests (@Test(timeout=))
- New Assertions in j.f.Assert Class
- No GUI Runner
- Use of suite() is Discouraged
- A New Runner – org.junit.runner.JUnitCore
  - > Based on Classes to Test
  - > No ANT Support Yet
-

JUnit4: Important Classes



MIX AND MATCH WONT WORK

JUnit: Integration and Reporting

- ANT and NetBeans
- The Optional Tasks: <junit> and <junitreport>
- Compare the JBoss Results ...
- NetBeans has Support for JUnit-3.x Tests (Plugin)
  - No Plugin for JUnit-4
-

JUnit: Some Objections

- Logging is not Built-in – Reports Have Linked Access (Log4Unit)
- Reactive Model to “Combat” TestNG (e.g @Ignore)
- The Test Class Invocation => No of “Tests”
  - > Too Much of “Static” Code
- Test Dependency can not be achieved
  - > “Server” Environments
  - > Success, Failure .... SKIP

TestNG: Philosophy

- (Try to) Provide What's not Present in JUnit
  - > The So-called JUnit Frustrations
- Unit Testing -> Integration Testing
- Pioneered Annotation Based Testing
  - > Lots of 'em
  - > Superset of What JUnit Has
- Talk of Suite, Test, Class
  - > <suite><test><classes>...
- Novel Way of Looking at Testing Java Programs?

TestNG: Features ...

- Annotations With a Close Control
  - @Configuration (before/after Suite/Test/Class/Method, groups, enabled, dependsOnMethods ...)
  - @DataProvider (name)
  - @ExpectedExpectations, @Factory
  - @Test (alwaysRun, dataProvider, timeOut, groups, enabled ...)
- Procedure
  - Write the Test – Import TestNG Annotations (No Ext..)
  - Write an xml file: testng.xml

TestNG: Features ...

- Procedure:
  - > Use ANT Task (Not Built-in into 1.6.2) (taskdef)
  - > Just Run the <testng> Task and Reports are Ready
- Richer Use of TestSuites, Tests, Groups, Packages, Classes
  - > Does Increase the Complexity
  - > Schema for XML is a Little Involved
  - > Arguably, a Closer Control Over What's Run
- XML's Benefits and Woes
  - > Explains Why the Guy is from EJB Development :)
  - > <include>, @Parameters

TestNG: Features ...

- @DataProvider
  - > @DataProvider(name="foo")
  - > @Test (dataprovider="foo")
  - > Object[][]
  - > Dimension Decides Test Iterations
- Dependent Methods – dependsOnMethods,Groups
- Factory of Test Classes
  - > A Different Way
- Can Run JUnit Tests
  - > Replace JUnit, NOW...

Comparison (Lot of Resources ...)

• Years in Business	JUnit
• Support	Both (testng.dev.java.net)
• Integration(IDE, Tools)	TestNG
• Simplicity	JUnit
• Learning Curve	JUnit
• Framework Adoption	Both (JUnit3/4 Dilemma?)
• Innovation	TestNG (Slightly)

My Recommendation -- TestNG

Thank You

Kedar Mhaswade
kedar.mhaswade@sun.com