

FLEX UNIT 4 AND HUDSON

Test Driven Development and Continuous Integration for
Actionscript



WHO AM I?

Who do you think you are Mr Winchester?

ASSUMPTIONS

- Version Control is essential and it is used in your development process
- Most code is not inside FLA's
- Ideally NO CODE is inside FLA's

VERSION CONTROL

- Enables reliable monitoring of changes to source code
- You know when code has changed
- What code has changed
- Who changed it (the finger of blame)

NO CODE IN FLA'S

- Version Control Systems work on text files.
- They can work with binary files but the incremental changes are not human readable.
- Code in FLA's is held in binary form and so changes can not be easily monitored.

RECOMMENDATIONS

- Subversion
- Git
- Mercurial
- Not CVS (old and outdated now)

THINGS I WILL BE USING

- Flash Builder
- Subversion
- Eclipse Subversive Plugin
- Ant
- Hudson

SO WHAT'S THE PROBLEM?

- Shit Happens!
- We want as little shit to happen as possible.



SOFTWARE HAS BUGS

- We want to minimise the number, and
- We want change to be safe
- How?

WE HAVE TO TEST!

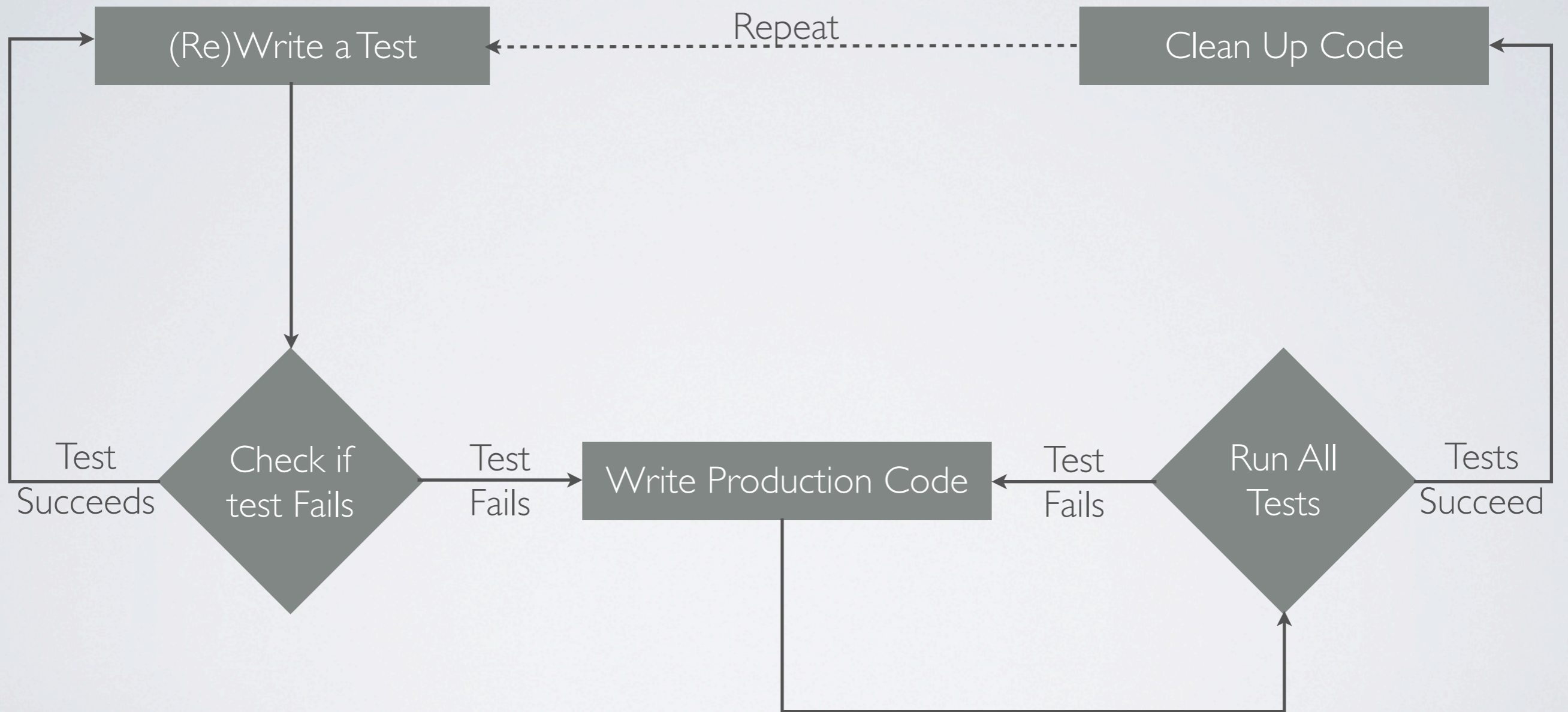
ENTER TDD

- Test Driven Development (TDD) is a strict discipline that combines
- Test First Development
- Refactoring
- It relies on Unit Tests

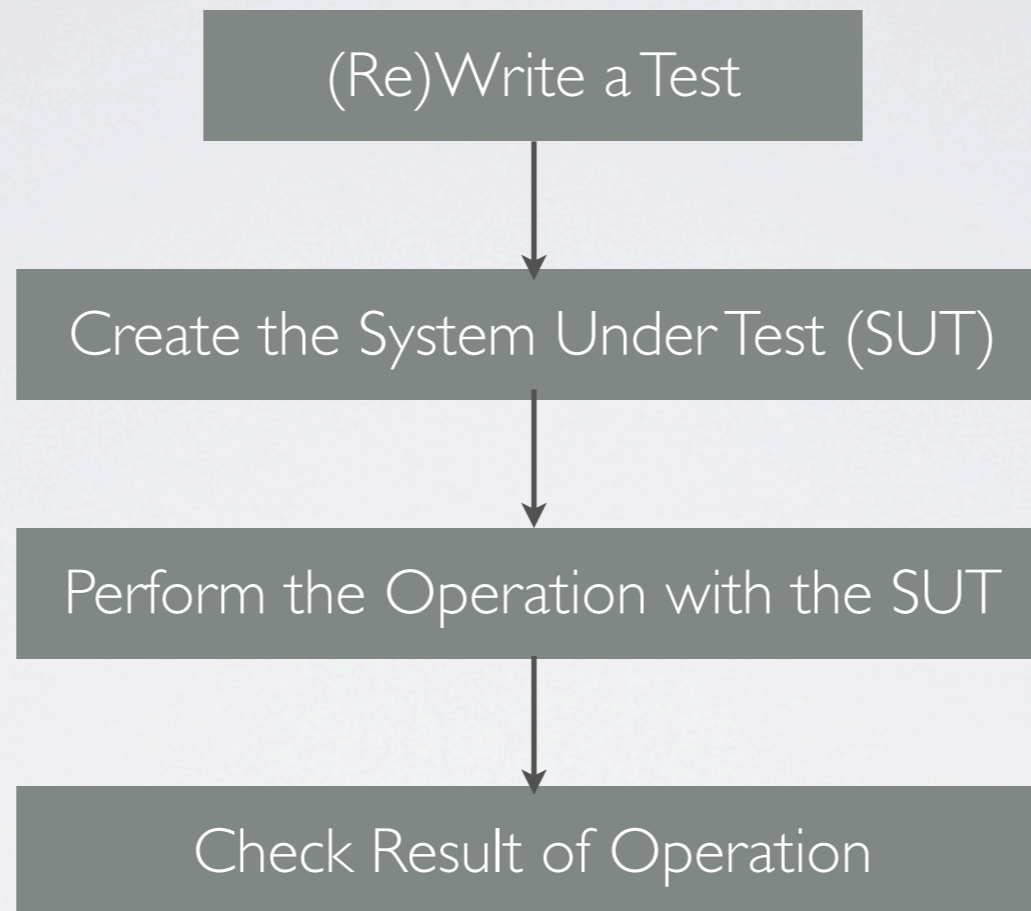
TEST FIRST DEVELOPMENT

- Write a test that fails (red light)
- Write some code to pass the test (green light)
- Refactor - making sure all tests pass

TDD WORKFLOW



WHAT IS A TEST LIKE?



WHAT IS A UNIT TEST?

- It is a software validation and verification method
- It checks that an individual *UNIT* of source code is fit for use
- Usually these *UNITS*s are at the level of method calls in the code
- Ideally each unit test is independent of all the others

WRITING SIMPLE TESTS

- Test each exposed method of a class
- Make sure tests exercise unexposed methods
- Avoid writing test code inside production classes

EXAMPLE SPECIFICATION

- Write a calculator class
- It must add numbers
- It must subtract numbers
- It must multiply numbers
- It must divide numbers

LET'S WRITE A TEST

ACTIONSCRIPT UNIT TESTING FRAMEWORKS

- AsUnit started in 2004
- FUnit is a new kid on the block
- Flex Unit (formally as3flexunitlib on google code) earliest code seems to be end of 2006
- Fluint (formally dpUnit) came from digital primates in early 2008

ASUNIT

- Seems to have a good following
- There is an AS2 version
- I don't know much about it
- More info at <http://asunit.org/>

FLEXUNIT 0.9

- Originally a google code project, but as of August 2008 it was adopted as the official unit testing framework for Flex by Adobe
- It is now maintained at opensource.adobe.com
- It's official version is 0.9, it never made it to 1.0
- The name is misleading, it is an AS3 testing framework and does not have any reliance on Flex libraries.

PROBLEMS WITH FLEXUNIT AND ASUNIT

- Both are good for unit testing code, if that is all you want to do.
- Both use an approach based on older version of JUnit (a java unit testing framework)
- Every class that contains a test must extend a base TestCase class
- No testing to catch generated exceptions
- Very bad at integrating with the flex component lifecycle

FLUINT

- Developed to deal with the problems of the other
- Extremely good at asynchronous tests
- Introduced a mechanism for 'faking' the component life cycle
- Good asynchronous test support
- Every class that contains a test still must extend a base TestCase class
- No testing to catch generated exceptions

FUNIT

- A modern xUnit style testing framework
- Similar to FlexUnit 4, but not as complete
- In particular no UI impersonation
- Not an adobe standard
- Syntax diverges from standards (like JUnit 4)

ANY MORE?

- There are more frameworks, BUT as far as I am aware none of them
- are as complete
- are as supported

INTRODUCING FLEX UNIT 4

- New kid on the block
- The bastard love child of Fluint and FlexUnit 0.9

FLEXUNIT 4

- xUnit compliant testing framework
- Uses metadata to define tests
- No need to extend a base class
- Has all the features of Fluint and then some!

ARCHITECTURE OF FLEX UNIT 4

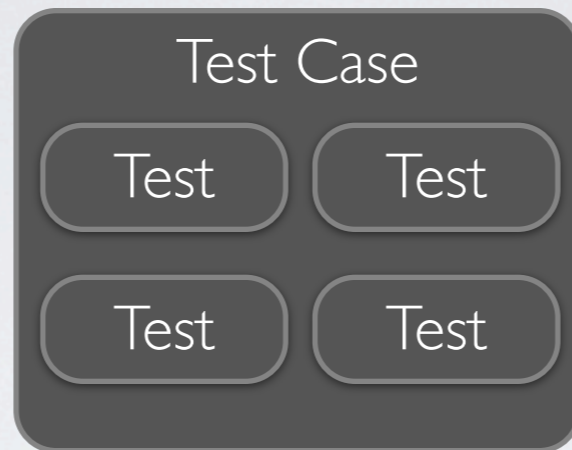
ARCHITECTURE OF FLEX UNIT 4

Test

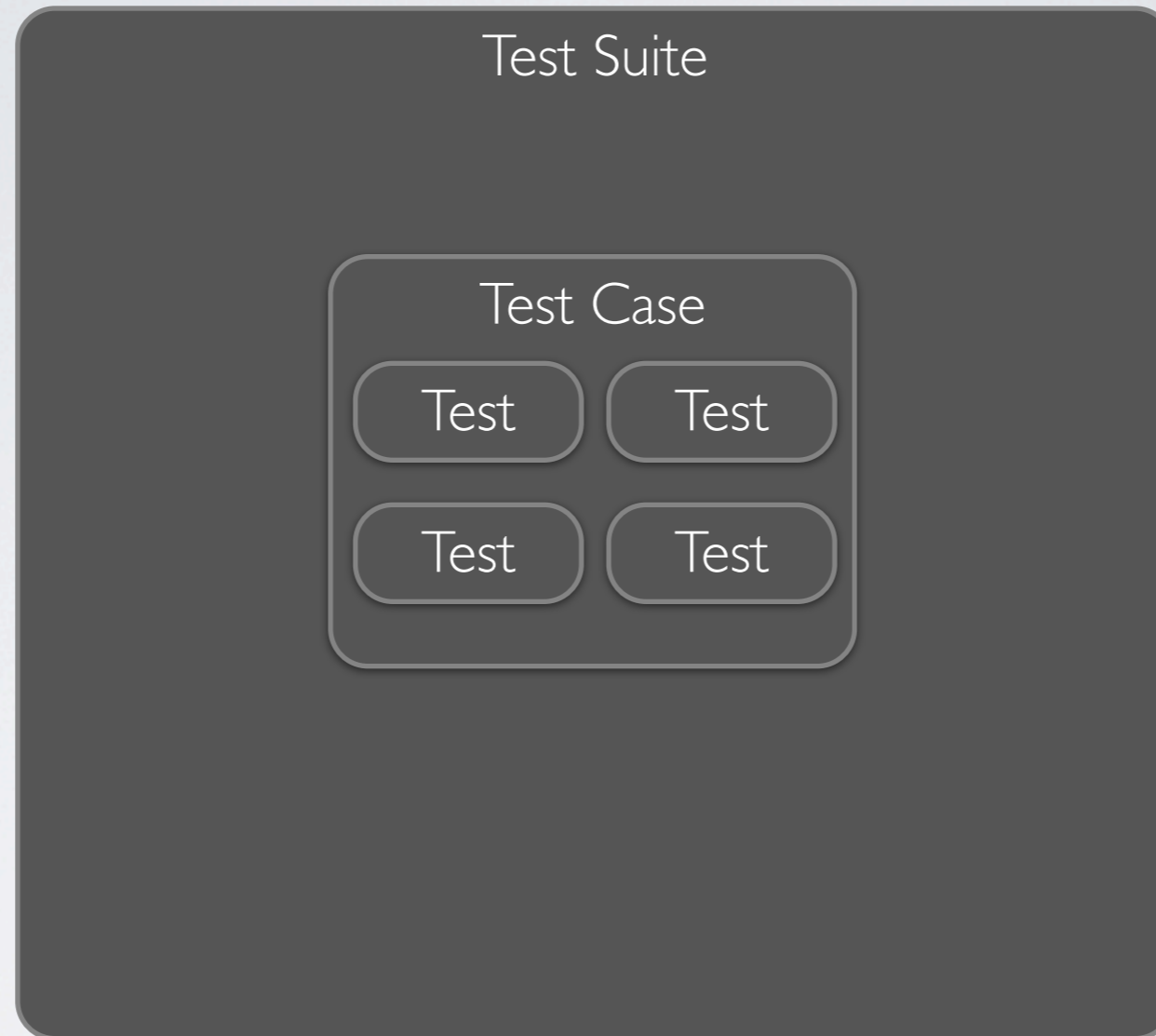
ARCHITECTURE OF FLEX UNIT 4



ARCHITECTURE OF FLEX UNIT 4



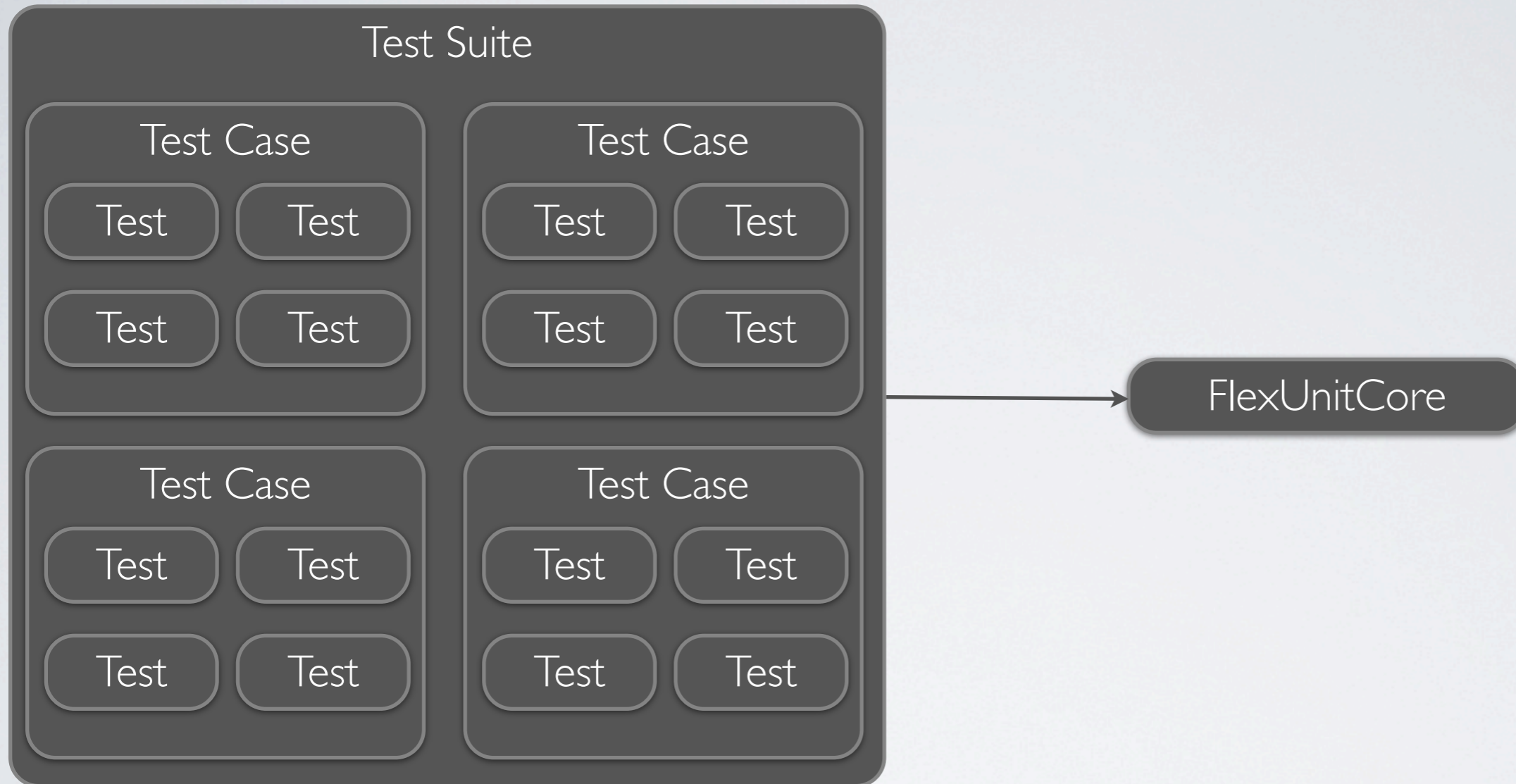
ARCHITECTURE OF FLEX UNIT 4



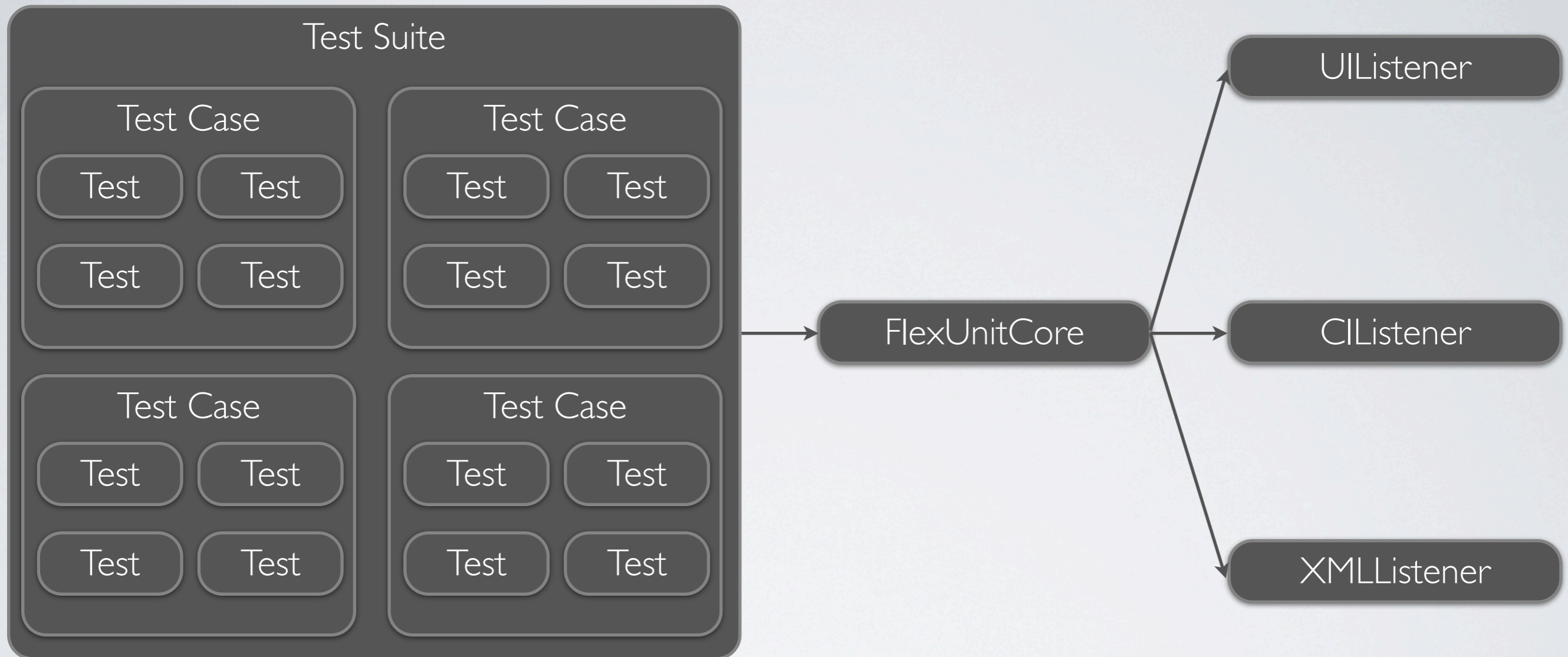
ARCHITECTURE OF FLEX UNIT 4



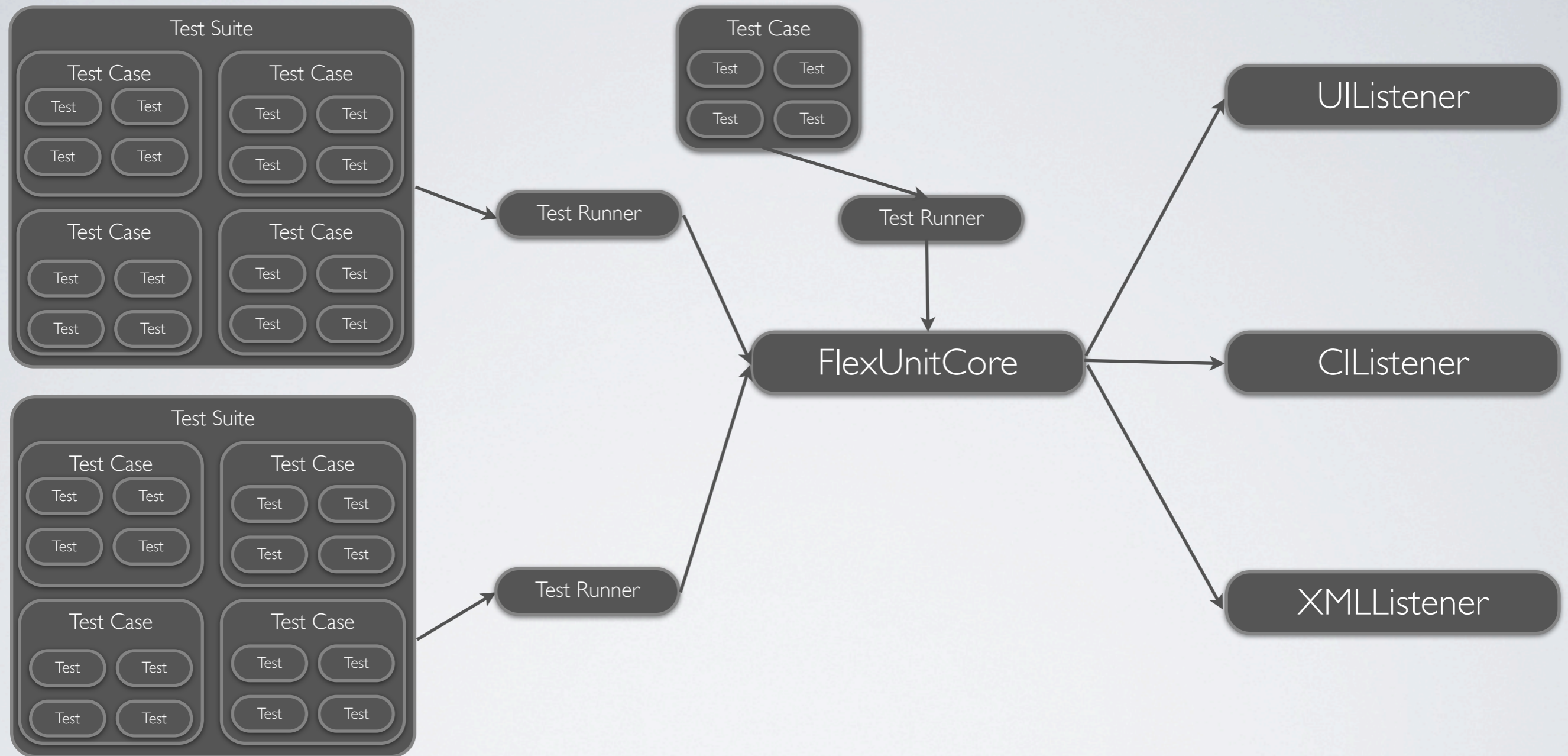
ARCHITECTURE OF FLEX UNIT 4



ARCHITECTURE OF FLEX UNIT 4



ARCHITECTURE OF FLEX UNIT 4



RUNNERS

- Flex Unit 0.9 runner
- Flex Unit 4 runner
- Fluint Runner
- ASMock Runner

LISTENERS

- UIApplication
- CLLocation
- XMLHttpRequest

A PROBLEM!

- Unless tests are run automatically, testing is ineffective.

ENTER ANT

- Industry standard build process
- A way to use mxmhc from the command line
- Enables us to run tests as part of an automated process

ANT TASKS

- flexTasks.jar - contains the tasks for building applications and libraries using mxmhc and compc
- FlexUnitTasks.jar - contains the tasks for getting the results of tests from a running flash application

ENTER HUDSON

- Will automatically run tests for you
- Will tell you if they fail
- Can be downloaded from: <http://hudson.dev.java.net/>
- Very easy to setup and configure

ADVANCED

- XVNC
- Plugins
- Fun Lights

SETTING UP HUDSON

- The lights

FLEX UNIT 4 - ADVANCED

- Asynchronous tests
- Using the UIImpersonator
- Special surprise - integration directly in Flash Builder

ADVANCED TESTING

- Mocking
- Stubs
- Dependency injection
- Coverage

CONCLUSION

- Testing is a very powerful tool to help produce, maintain and impart confidence in your software.
- Actionscript 3 now has testing tools approaching the level available to Java developers
- We should all be testing!