

The one bit everyone forgets: Testing your ADF application with JUnit



John Stegeman
Xchanging



<http://groups.google.com/group/adf-methodology>

ADF EMG



- A place to discuss **best practices** and methodologies for JDeveloper ADF enterprise applications
- Founded mid-2008 by Chris Muir, now **600+ members**
- Focus is **Fusion Tech Stack** (ADF Faces, ADF BC)
- Online forum plus sessions at major Oracle conferences (OOW, ODTUG, UKOUG, DOAG...)

About me

- 12 years using JDeveloper (since v3)
- 21 years using Oracle DB + tools
- Oracle ACE Director (Fusion Middleware)



John Stegeman
Xchanging



Agenda

- Testing / JUnit concepts
- Using JUnit with JDeveloper – a simple example
- Testing ADF applications with JUnit
- What to test and what *not* to test
- Resources



John Stegeman
Xchanging



TESTING AND JUNIT CONCEPTS



John Stegeman
Xchanging



Why test?



John Stegeman
Xchanging



Some types of software testing

- Unit
- Integration
- System
- System Integration
- Non-Functional (performance, scalability, usability, etc.)



John Stegeman
Xchanging



The importance of early testing

Cost to fix a defect		Time detected				
		Req'ments	Architecture	Constr'ion	System test	Post-release
Time introduced	Requirements	1x	3x	5–10x	10x	10–100x
	Architecture	-	1x	10x	15x	25–100x
	Construction	-	-	1x	10x	10–25x

Source: http://en.wikipedia.org/wiki/Software_testing



John Stegeman
Xchanging



Early testing is important, so why don't we do it well?

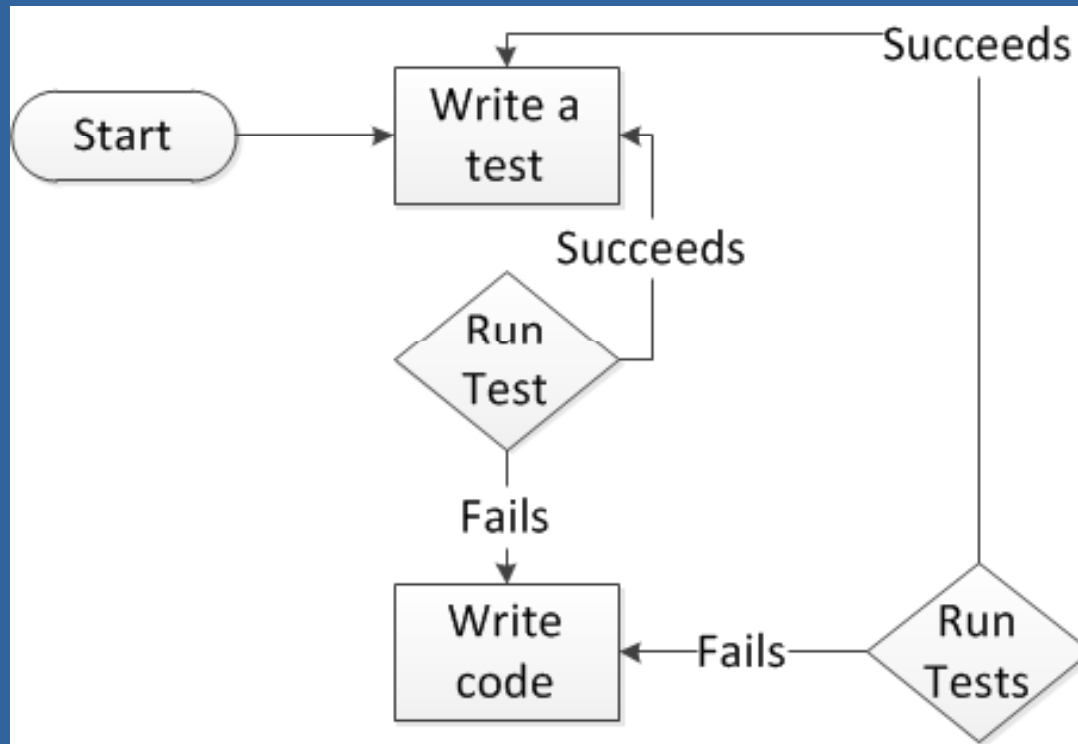
- It takes too much time
- Too much manual work
- Unit testing treated as a “one-off” process
- No experience with tools
- Never taught how
- Hard to get started
- Fill in your reason here: _____



John Stegeman
Xchanging



Test-Driven Development (TDD)*



* Can be challenging with ADF apps!

What is JUnit?

- One of the so-called “xUnit” frameworks
- Framework for writing and running automated unit tests in Java
- Can be part of a plan to overcome unit testing challenges
- Can help enable TDD



John Stegeman
Xchanging



Some JUnit Terminology

- Test Method
- Test Class
- Assertion
- Test Fixture
- Test Suite



John Stegeman
Xchanging



USING JUNIT WITH JDEVELOPER



John Stegeman
Xchanging



Demonstration

- Enabling JUnit for JDeveloper
- TDD for a “factorial” class



John Stegeman
Xchanging



Factorial specifications

- $0! = 1$
- For any positive integer N , return $N!$ (the product of all positive integers less than or equal to N)
- For any negative integer, throw an `ArithmeticException`

TESTING ADF APPLICATIONS



John Stegeman
Xchanging



Testing ADF Applications

- “Classic” unit tests are difficult to achieve
 - External dependencies (such as databases)
 - Code cannot easily be executed in isolation (JSF, EO, VO)
- This doesn’t mean you should give up!
- JDeveloper Extension for BC testing – useful for generating boilerplate



John Stegeman
Xchanging



Testing ADF Business Components

- Create AM configurations to use specifically for your tests
- Tests will start from an AM – use the extension
- Difficult to directly test EOs



John Stegeman
Xchanging



Testing JSF (ADF Faces) apps

- Impossible (or nearly so) to unit test in the *classic* sense
 - Lots of bits of code that don't stand alone
 - Dependent on container (or extensive set of “mock objects”)
 - Dynamically instantiated classes from ADF Page Definitions
 - Usually require external resources (such as databases)



John Stegeman
Xchanging



Testing JSF (ADF Faces) apps

- Instead of unit tests, you can do integration or system tests
- JUnit-compatible frameworks/tools:
 - Selenium (create UI-level tests)
 - JSFUnit (can test internal JSF state as well – not personally tested with ADF)
- Will generally require deploying your app to an application server



John Stegeman
Xchanging



WHAT TO TEST AND WHAT *NOT* TO TEST



John Stegeman
Xchanging



What to (and not to) test?

- Test code that *you* write (this includes “code that isn’t really code”)
- Don’t test code that *Oracle* writes, except:
 - when you implement a workaround for an Oracle bug
 - when you expect behaviour might change in the future (release notes, etc.)



John Stegeman
Xchanging



Practical suggestions – ADF Business Components

- Validation rules
- Default values
- Framework Extension classes (test by checking the behaviour of classes that inherit)
- Application Module service methods
- Any code that you write in EO/VO Java classes



John Stegeman
Xchanging



Practical suggestions – ADF Faces

- Use Selenium to test the UI
- Focus on the displayed output and navigation
- Test database interaction that you haven't already tested in the ADF BC tests



John Stegeman
Xchanging



Resources

- JUnit homepage: <http://junit.org>
- Some basic docs: <http://junit.sourceforge.net>
- ADF Development Essentials (parts 5,6): <http://www.oracle.com/technetwork/articles/adf/adf-essentials-098792.html>
- EMG: <http://groups.google.com/group/adf-methodology>



John Stegeman
Xchanging



Thank you for listening!



John Stegeman
Xchanging

