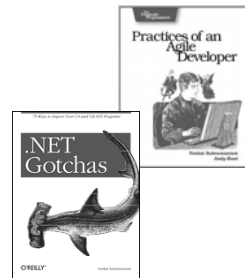


Open Source Tools for Agile Development

Venkat Subramaniam

```
spkr.name = 'Dr. Venkat Subramaniam'  
spkr.founder = 'Agile Developer, Inc.'  
spkr.affiliated = 'University of Houston'  
spkr.associated = 'Rice Univ. Continuing Studies'  
  
spkr.cred = %w{Programmer Speaker Trainer Author}  
  
spkr.nfjs = 'NFJS Speaker since 2002'  
  
spkr.website = 'http://www.agiledeveloper.com'  
spkr.email = 'venkats@agiledeveloper.com'
```



Abstract

Several tools play an important role in agility and productivity. What tools will benefit developers? Instead of going through a laundry list of tools available, we'll discuss, using examples, their practical use.

We'll start by looking at tools for unit testing and mock objects. Then take a look at tools that will help ensure certain performance of critical code.

You know writing good code is more than simply using an OO language. We'll look at tools that will help with code metrics, so you can analyze, and refactor your code to reduce coupling and undesirable dependencies.

But, what about hidden critical errors in your code, like synchronization problems that may potentially lead to deadlocks? We'll look at open source tools to proactively eliminate these from your code.

Finally, we'll look at tools for automation and extreme feedback though out the development cycle.

Open Source Tools for Agile Dev.

- Why interest in tools?
- Development Tools
- Testing Tools
- Performance Tools
- Automation Tools
- Project Management Tools
- Modeling Tools
- Conclusion

Core values of Agile Development

- Agility
- Communication
- Continuous progress
- Feedback
- Working software is measure of progress

Agile Development and Tools!

- First value in Agile Manifesto:
Individuals and interaction over process and tools
- Agility depends on you – the person that makes the team, and your team
- “A fool with a tool is a dangerous fool”
- Good tools help you succeed, but can’t promise success
- A good developer needs effective tools to be productive and succeed

Open Source Tools for Agile Dev.

- Why interest in tools?
- Development Tools
- Testing Tools
- Performance Tools
- Automation Tools
- Project Management Tools
- Modeling Tools
- Conclusion

Refactoring

- Refactoring
 - One of the key features of agile development
- TDD Mantra: Red – Green – Refactor
- “A process of changing a software system in such a way that it does not alter the external behavior of the code yet improves its internal structure”
- Several refactoring techniques exist
- We rely on IDEs for this
 - Eclipse IDE

Unit Testing

- Black box testing vs. White box testing
- Two kinds of feedback
 - Code meets developers expectations?
 - Unit Testing helps with this
 - Act of design than verification
 - Serves as a safety net
 - Is a form of documentation
 - Code meets customers expectations?
- Creating test cases
- Automating execution
- Categorizing tests

JUnit

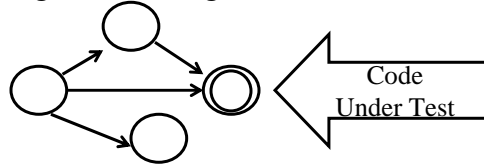
- The framework that popularized unit testing in Java community
- You inherit from TestCase
- Methods prefixed by "test"
- GUI and text based output

TestNG

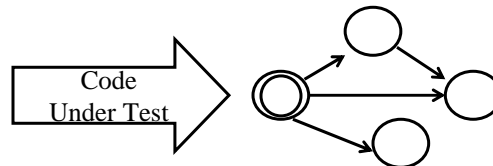
- Testing, the Next Generation
- JUnit has some limitations
 - For instance, need to inherit from TestCase
- TestNG supports annotation style
- Data driven testing
- Test distribution

Where UT gets hard

- Unit Testing is easy when your code has no dependencies



- It gets hard when we have dependencies
 - Leads to finding excuses not to UT



- Mock objects come to rescue

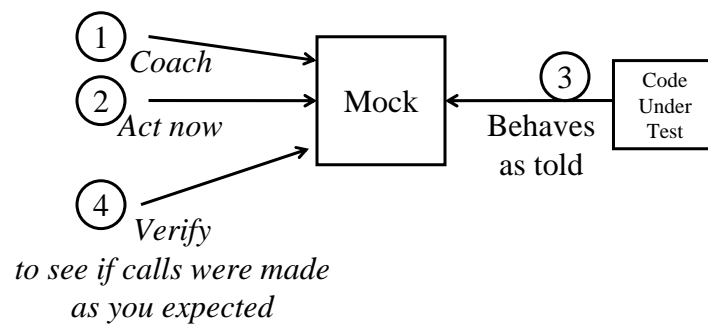
Mock Objects

- A Mock object stands in for the real object
- It is an object that you first coach... tell him what to say and how to dance, then you set him loose, and he mimics what you coached
- Simulates the behavior of your dependencies

Tools for Mock

- You can hand toss mocks, but can get tedious
- EasyMock
- JMock
- MockRunner – mocks containers, etc...

EasyMock



Stable Dependency Principle

"Dependencies between released components must run in the direction of stability. The dependee must be more stable than the depender."

"Agile Software Development – Principle, Practices, and Patterns" by Robert Martin

Stable Dependency Principle

- A component can never be more stable than the one it depends upon
- Instability $I = C_e / (C_a + C_e)$,

where

C_a - # of classes outside that depend upon this class

C_e - # of classes outside that this class depends upon

- $0 \leq I \leq 1$
- **0 - ultimately stable; 1 - ultimately unstable**

Stable Dependency Principle...

Components should be arranged such that components with a high I metrics should depend upon component with low I metrics

Stable Abstraction Principle

"The more stable a component is, the more it must consist of abstract classes. A completely stable category should consist of nothing but abstract classes."

Stable Abstraction Principle

- Implementation of methods change more often than the interface
- Interfaces have more intrinsic stability than executable code

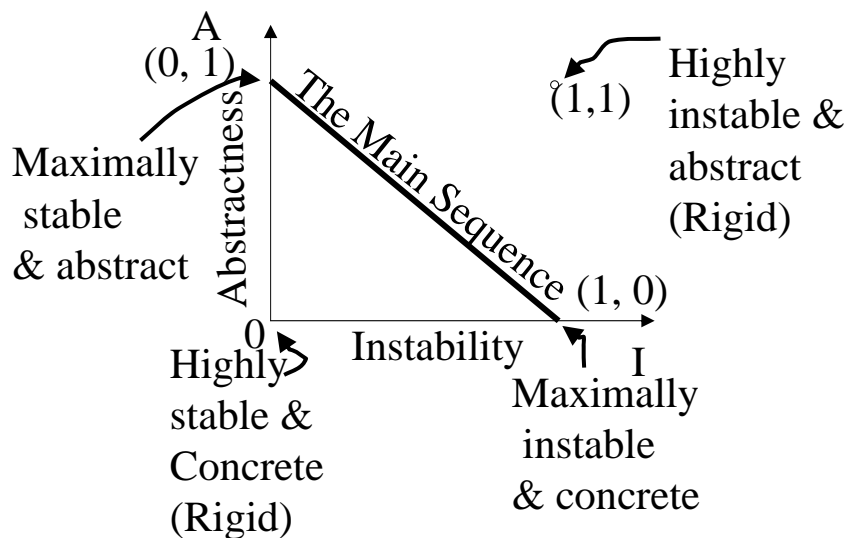
- Abstraction of a Component

$$A = (\# \text{ of abstract classes}) / (\# \text{ of classes})$$

$$0 \leq A \leq 1$$

- 0 - no abstract classes; 1 - all abstract classes

Stability Vs. Abstractness



Distance from the main sequence

- $D = |(A + I - 1) / \sqrt{2}|$
- $0 \leq D \leq 0.707$; *Desirable value of D is closed to 0*
- Normalized form $D' = |(A + I - 1)|$
- Calculate D value for each component
- Component whose D value is not near Zero can be reexamined and restructured

Tool to Report Dependency

- How about figuring out the metrics from your code
- JDepend
- You can determine these metrics
- Or assert it as part of your unit test!
- JDepend4Eclipse – a plug-in for Eclipse

Coverage

- Unit Testing is only as good as the tests you write
- If unit tests don't touch your code, what good is it
- If you follow test first coding, you may have 100% code coverage, but...
 - That is hard. We may throw in some conditions, but did we reach all those branches?
- JCoverage presents details about how much of your code is covered
- Cobertura is a code coverage graphing application

Test your Test

- JUnit test tester –Jester – is a tool that will tell you how good your tests are
- What coverage does not do?
 - Tells you if a line of code is executed, but...
 - Does not tell you if it executed correctly
- Jester will modify code to see if the test fails
- It proactively modifies code to see if you have considered different input or combination of input
 - May report some false positives, however

Simian

- The DRY Principles
- One of the most common mistakes
 - We copy and paste naturally ☺
 - Results in increased cost
 - hard to track problems
 - hard to maintain codes
- Simian – Similarity Analyzer – Finds duplication in code
- PMD's Copy/Paste Detector (CPD)

Bugging the bug

- Hidden bugs in code
- Sooner we find, the better it is
- FindBugs is a open source tool to detect bugs in Java program
- Tabularizes errors and warnings with good information

Open Source Tools for Agile Dev.

- Why interest in tools?
- Development Tools
- Testing Tools
- Performance Tools
- Automation Tools
- Project Management Tools
- Modeling Tools
- Conclusion

Beyond Unit Testing

- What about testing UI
- What about functional, integration testing
- What tools can help us with that?

Testing UI

- Testing UI is not easy
 - OK it is a pain in the... neck
- Often people confuse unit testing with acceptance and usability testing when it comes to UI
- Jemmy
- Abbot
- jfcUnit

Framework for Integration Testing

- FIT
 - Collaboration tool
 - you learn what software should do
 - then automatically compare that to what it actually does
- FitNess - HTML and wiki frontend to FIT
- You can express the expectations as tables

Testing Web Applications

- How do you test behavior of a web application
- Need to handle HTML, session information, etc.
- HttpUnit
- Selenium
- Watir

Open Source Tools for Agile Dev.

- Why interest in tools?
- Development Tools
- Testing Tools
- Performance Tools
- Automation Tools
- Project Management Tools
- Modeling Tools
- Conclusion

Performance

- “Premature optimization is the root of all evil” – C.A.R. Hoare
- Need to focus where the problem is
- But where’s the problem
- Tools that help you zero in on the problem

JUnitPerf

- Extension to JUnit
- Allows you to time bound a test
- If test is failure if it does not finish within allocated time
- Load test available as well

JMeter

- Load test functional behavior
- Measure performance
- Works on POJO, servlet, ...
- Simulates load on service

JLint

- Very hard to find synchronization problems
- Run it 70 times, you still can't find it
- Best to analyze and understand the code
- That's where JLint comes in
 - Finds bugs, inconsistencies, synchronization problems

Open Source Tools for Agile Dev.

- Why interest in tools?
- Development Tools
- Testing Tools
- Performance Tools
- Automation Tools
- Project Management Tools
- Modeling Tools
- Conclusion

Automation tools

- No one like to be that build monkey on projects
- Yes, some of us fall prey
- Consider automating the build and more important running that unit test – continuous integration
- Ant, Maven
- CruiseControl, AntHill, DamageControl,...

Open Source Tools for Agile Dev.

- Why interest in tools?
- Development Tools
- Testing Tools
- Performance Tools
- Automation Tools
- Project Management Tools
- Modeling Tools
- Conclusion

Knowing where things stand

- Managing project
- Bug tracking, schedule tracking...
- Bugzilla
- Tack+
- XPlanner

Open Source Tools for Agile Dev.

- Why interest in tools?
- Development Tools
- Testing Tools
- Performance Tools
- Automation Tools
- Project Management Tools
- Modeling Tools
- Conclusion

Modeling

- Modeling is a critical step in agile development
- Modeling is not documentation
- Modeling is thinking about design
- Documentation is writing about your design
- Tools can help represent your ideas – UML
 - ArgoUML
 - Jude
 - Violet – plugs in to Eclipse

Quiz Time



Open Source Tools for Agile Dev.

- Why interest in tools?
- Development Tools
- Testing Tools
- Performance Tools
- Automation Tools
- Project Management Tools
- Modeling Tools
- Conclusion

Conclusion

- Agile development depends on communication and feedback
- Tools can help us get productive
- We saw different tools to achieve this
- There are lot more tools
- Find the one that meet your need

Tools List

Abbot <http://abbot.sourceforge.net>
Ant <http://ant.apache.org>
ArgoUML <http://argouml.tigris.org>
Bugzilla <http://www.bugzilla.org>
Track+ <http://www.trackplus.de>
Code Analyzers <http://java-source.net/open-source/code-analyzers>
Continuous Integration tools
<http://damagecontrol.codehaus.org/Continuous+Integration+Server+Feature+Matrix>
Cobertura <http://cobertura.sourceforge.net>
Cruise Control <http://cruisecontrol.sourceforge.net>
Easy Mock <http://www.easymock.org>
Eclipse <http://www.eclipse.org>
FindBug <http://findbugs.sourceforge.net/commons-modeler.html>
FIT <http://fit.c2.com>
FitNess <http://fitnesse.org>
HttpUnit <http://httpunit.sourceforge.net>
JCoverage <http://www.jcoverage.com>
JDepend <http://clarkware.com/software/JDepend.html>
JDepend4Eclipse <http://andrei.gmxhome.de/jdepend4eclipse>
Jemmy <http://jemmy.netbeans.org>

Tools List...

Jester <http://jester.sourceforge.net>
jfcUnit <http://sourceforge.net/projects/jfcunit>
JLint <http://artho.com/jlint>
JMeter <http://jakarta.apache.org/jmeter>
JMock <http://www.jmock.org>
Jude <http://objectclub.esm.co.jp/Jude>
JUnit <http://www.junit.org>
JUnitPerf <http://www.clarkware.com/software/JUnitPerf.html>
Maven <http://maven.apache.org>
MockRunner <http://mockrunner.sourceforge.net>
Open Source Software in Java <http://java-source.net>
PMD's Copy/Paste Detector (CPD) <http://pmd.sourceforge.net/cpd.html>
Project Management Tools <http://java-source.net/open-source/project-management>
Selenium <http://www.openqa.org/selenium>
Simian <http://www.redhillconsulting.com.au/products/simian>
TestNG <http://testng.org>
Violet <http://horstmann.com/violet>
Watir <http://wtr.rubyforge.org>
XPlanner <http://www.xplanner.org>

References...

You can find evolving list of tools
mentioned in this talk
(plus some more) at

<http://tinyurl.com/b5krx>

Download examples/slides from

<http://www.agiledeveloper.com/download.aspx>

Please fill out your evaluations!