RejmiNet

Testing Java Code: Beyond the IDE

Ian F. Darwin
RejmiNet Group Inc.
http://www.rejmi.net/

Session ID TS-9667

Goal of This Talk

What You Will Learn

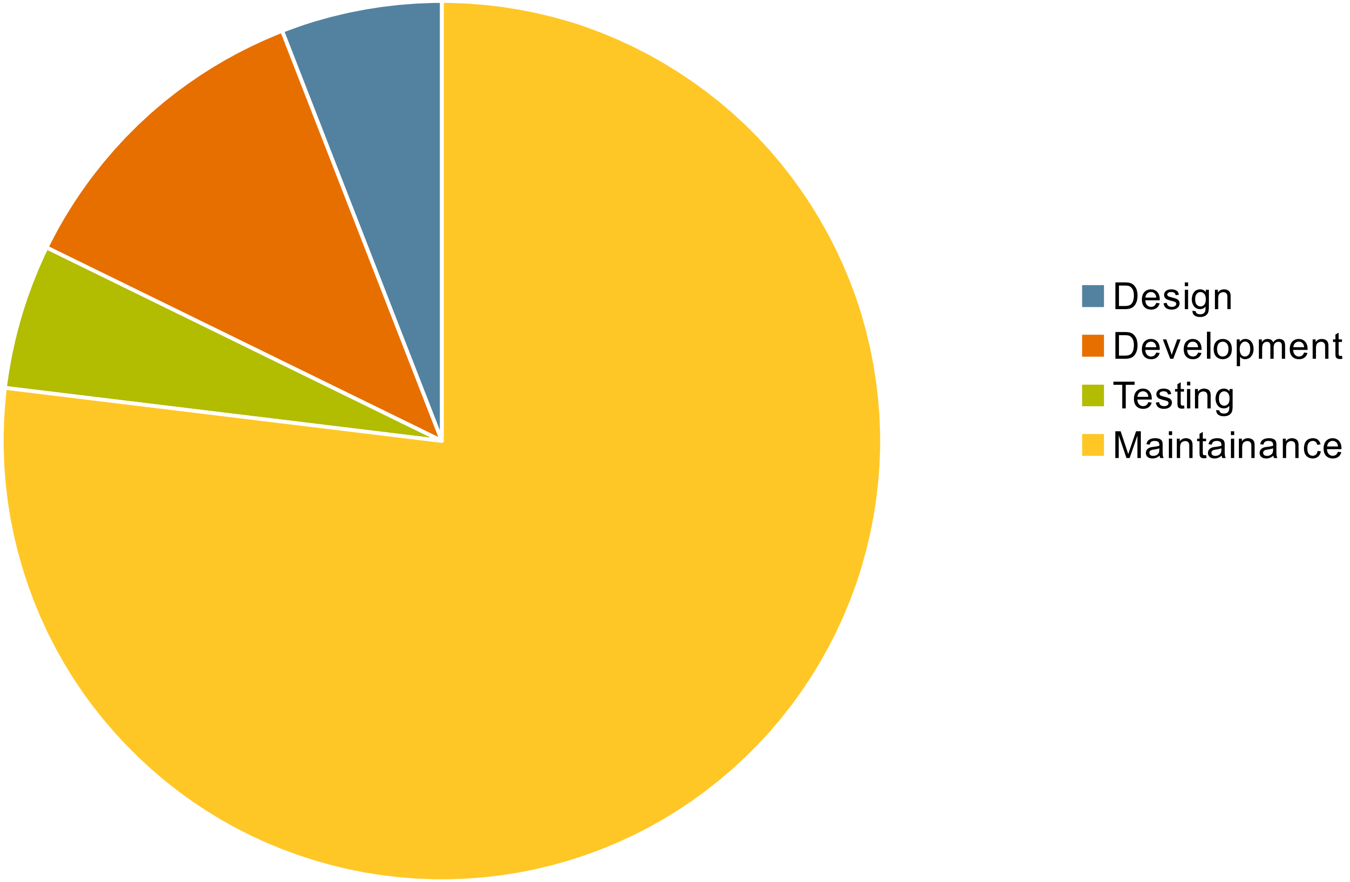
Learn to *improve Java program correctness* in one session!

Beyond checking that JDK/IDEs do

Using free, open source tools

Partly drawn from my new O'Reilly ebook *Checking Java Programs*
http://www.oreilly.com/catalog/9780596510237

Cost of Software Lifecycle (Average)



Averaged from several sources; see especially http://www.cs.jyu.fi/~koskinen/smcosts.htm

Why More Checking?

Because we can, and should

- Java's original goals included reliability
    - Why there is as much checking as there is
    - Garbage collection, threading support, ...
    - Compile time and run time checking
- But it's never enough
    - Software still has bugs, right?
    - Standard tools do not look at patterns, nor at many potential sources of failure
    - So we bring in extra tools
    - Must couple with developer education!

So What's Wrong With This Code?

```
// Part of SaveAction.actionPerformed()
if (fileName != null && !doingSaveAs) {
    doSave(fileName);
    return;
}
int returnVal = chooser.sho
if (returnVal == JFileChoos
    File file = chooser.getS
    if (file.exists() && doi
        int ret = JOptionPane                      ame,
              "File exists, overwrite?", "Overwrite?",
              JOptionPane.YES_NO_OPTION);
        if (ret != 0);      // "Yes" is the 0th option...
              return;
    }
    doSave(file);
}
```

Developers can spend up to 50% of their time understanding code before they can fix it...

Error here:
file does not get saved!

What to do?

My “big three” add-on tools

- PMD
  - Source code analyzer
- FindBugs
  - Class file analyzer
- NASA Java Pathfinder
  - Runtime state verifier
- And a few more for good measure

Agenda

PMD

FindBugs

Java PathFinder

Other

PMD

- PMD reads source code looking for patterns
- Performs its own parsing to an AST (tree)
  - Not a full compile, so can survive e.g missing imports
  - Tests match patterns in the AST
  - Extensible via XPath or Java code
- Can be run standalone, with Eclipse or NetBeans, Ant or Maven, and other tools (even emacs!)

Like the name Java, PMD is not an acronym.

Running PMD

Extending PMD

PMD has lots of rules already

Easy to add your own in XPath or Java

Has visual tool for exploring AST from code bits

Eg., ban library code throwing SQLException

```
//MethodDeclaration/NameList/Name[@Image='SQLException']
```

Embed in a 60-line XML file

- Download CJP book example (see last page)

PMD's Weapon of Mess Detection: CPD

- Copy-And-Paste bloats code, prevents reuse
  - Easy to commit, hard to ferret out
- CPD finds it, even with variable name changes
- Example from JDK:

```
Found a 294 line (531 tokens) duplication:
Starting at line 486 of src/java/lang/StrictMath.java
Starting at line 575 of src/java/lang/Math.java
    public static int round(float a) {
        return (int)floor(a + 0.5f);
    }
...etc...
```

Running CPD

- Command line or Ant task
- Specify min tokens
  - Too low matches getters
  - Too high causes false negatives
  - Start around 100 to avoid getter/setter

CPD Details

- Third implementation, uses Karp-Rabin algorithm
  - Does JDK classes source in under 5 seconds
- Works on Java, JSP, C, C++, and PHP code

DEMO

PMD

PMD

FindBugs

Java PathFinder

Other

FindBugs

From University of Maryland

Checks .class files (using Apache BCEL)

Can be used standalone or with Eclipse

Running FindBugs

- Note: this page is incomplete; I must have finished it at the conference and not committed it. Sorry. Please refer to my newer course on Testing Java Programs at O’Reilly (see https://cjp.darwinsys.com/ for details and links).

DEMO

FindBugs

Interlude: PMD and FindBugs

- PMD checks source code, FindBugs checks compiled .class files
- Overlap in what they find is only about 50%
  - Worth running both

PMD

FindBugs

Java PathFinder

Other

Java PathFinder

- Developed at NASA for testing software used in rocket control systems
  - No margin for error!
- Dynamic state checker
  - Can test all paths through code by “backing up and trying again”
- Includes its own JVM implemented in Java

JPF (continued)

- Good news:
  - Can find threading deadlocks
  - Can find “race conditions”
- Bad news:
  - JVM interface does not yet support AWT or sockets
    - Rockets don't have a GUI :-)

Agenda

PMD

FindBugs

Java PathFinder

Pushing the IDE

Other

Pushing the IDE

- Even without these tools, Eclipse, NetBeans can be made to do considerable checking
  - Eclipse can save these extra settings to the project

JUnit

“Never... have so many owed so much to so few lines of code”

JUnit is *the* best-known Unit Testing framework for Java; every coder should use it

I would love to talk about it for an hour, or for two days

If you don't already use JUnit, download it before you leave the room (http://www.junit.org/) and order a copy of J.B. Rainsberger's *JUnit In Action*

Just do it!

Churchill paraphrase is by Martin Fowler

No Shortage of Other Tools

- Jlint
  - Venerable code analyzer
- VerifyDesign
  - checks that code only uses allowed type
  - Supporting program-to-interface
- Jikes
  - IBM open source compiler with lots of warnings
- More!
  - See http://pmd.sourceforge.net/similar-projects.html

Summary

PMD: Static source code checker

FindBugs: Static byte-code checker

Java Pathfinder: dynamic model checker

May the Source be with you.
But not the bugs!

For More Information

See also:

- Other sessions: TS-2007 (FindBugs' author)
- Vast literature on software quality!
- My book: ***Checking Java Programs*** (O'Reilly “Short Cut” series)
  - See http://cjp.darwinsys.com/ for examples download
  - http://www.oreilly.com/catalog/9780596510237
- PMD: http://pmd.sourceforge.net/
- FindBugs: http://findbugs.sourceforge.net/
- JPF: http://javapathfinder.sourceforge.net/
- JUnit: http://www.junit.org/

Q&A

Ian Darwin
http://cjp.darwinsys.com/

Default Color Palette

Primary Colors

■	Sun Blue
■	Sun Orange
■	Sun Green
■	Sun Yellow
■	Sun Gray

Secondary Colors

■	Sun Dark Blue
■	Sun Dark Orange
■	Sun Dark Green
■	Sun Dark Yellow
■	Sun Light Blue
■	Sun Light Orange
■	Sun Light Yellow

The Three Main Tools Compared

Matrix Title: Use this matrix to input your data

	Q2 ’03	% to Q2 ’02
Units	Testing	+1%
Revenue	$1,000	+1%
Gross margin	1%	+1.0 points
OPEX	$1	+1%
Net income	$1	+1%
EPS	$.01	+1%

Source: Please add the source of your data here