Selecting and Managing the Use of Tools in an (undergraduate) Software Engineering Course

Jonathan Cook, NMSU

SEES @ ACM FSE

November 12, 2012

The Times They Are A'Changin'

WHEN I WAS A KID, COMPUTER PROGRAMS WERE MADE UP OF HUGE STACKS OF PERFORATED CARDS.
IN ORDER TO CREATE A SIMPLE PROGRAM, THOUSANDS OF THESE CARDS HAD TO BE PUNCHED AND PAINSTAKINGLY ARRANGED IN EXACT SEQUENCE!
THAT'S WHEN NERDS WERE NERDS, MY FRIEND!
WHAT CENTURY DID YOU SAY YOU WERE FROM?
SCOTT AND BORGMAN

By Jerry Scott & Jim Borgman
© 2012 ZITS Partnership

There's never been a better time to be a software engineering instructor!

(and hopefully it will continue to get better) http://www.laughitout.com/2010/01/also-bridge-is-out-ahead.html

CAUTION
THIS SIGN HAS
SHARP EDGES
DO NOT TOUCH THE EDGES OF THIS SIGN
ALSO, THE BRIDGE IS OUT AHEAD

The content of this talk is the opinion of the speaker and does not necessarily represent those of the ACM, FSE, SEES, NMSU, NSF, DOD, UFOs, ...

I use and advocate for FOSS, and this is where I'll spend most of my time today

New Mexico State University CS

- 12 faculty, 1 instructor, 230 ugrads, 100g
- BS/MS/PhD, new BA, minors, no tracks
- One undergrad software engineering course, plus a senior project course
  - CS 371, Software Development
- Expose students to broad overview of SE, plus team and tool exposure

What makes a good course tool?

- Focuses on learning concepts rather than peculiarities
  - Use is “straightforward”
- Available or **very** easy to install
  - Am not going to spend my time installing lots of dependencies and doing tricky configuration
  - Prefer to have tools that students themselves can install (assignment: “go get tool X...”)
- Benefits should be “obvious”

GET US SOME RISK MANAGEMENT SOFTWARE.
scottadams@aol.com
www.dilbert.com
WHAT CAN RISK MANAGEMENT SOFTWARE TELL YOU THAT MY COMMON SENSE AND EXPERIENCE CAN'T?
© 2009 Scott Adams, Inc./Dist. by UFS, Inc.
3-16-09
DATA.
STOP FAILING THE TURING TEST!

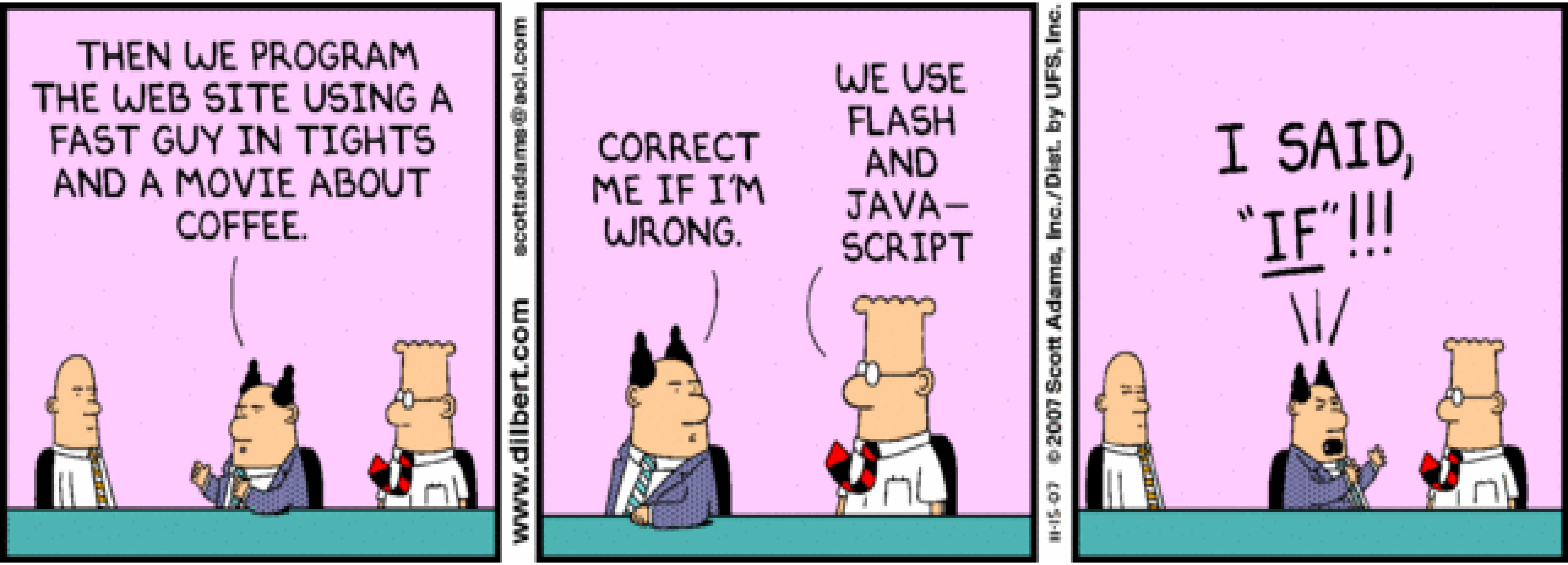
Tools List

- Language: Classic / novel / niche languages
- Library: Languages come with BIG libraries
- Framework: Design constraining application frameworks (Ruby/Rails/MVC)
- IDE: Integrated Development Environment
- Repository: Revision Control Systems
- Issue tracking and Wiki tools
- Build: Automated derived-product construction
- Code documentation: Comment-processing code documentation
- Test: Automated unit testing, coverage analysis, many others
- Diagramming: General, UML, brainstorming
- Verification (JML, EscJava, Forge, FindBugs, Jpathfinder, JLint)
- LMS tools

Tool Complexities

- Many tools target “all developers”
  - Try to be “industrial strength”, scalable
  - E.g., I won't use tools that need mysql
- I need to know “everything”
  - Not just click-install
- Students need to know “everything”
  - Not just click-install
- We are supposed to be the experts!

I don't want to be a sysadmin, but I do want to be competent in tool/platform use, and I want my students to begin being competent


THEN WE PROGRAM THE WEB SITE USING A FAST GUY IN TIGHTS AND A MOVIE ABOUT COFFEE.
scottadams@aol.com
www.dilbert.com
CORRECT ME IF I'M WRONG.
WE USE FLASH AND JAVA-SCRIPT
© 2007 Scott Adams, Inc./Dist. by UFS, Inc.
11-15-07
I SAID, "IF"!!!

So:

Easy to install

Easy to understand

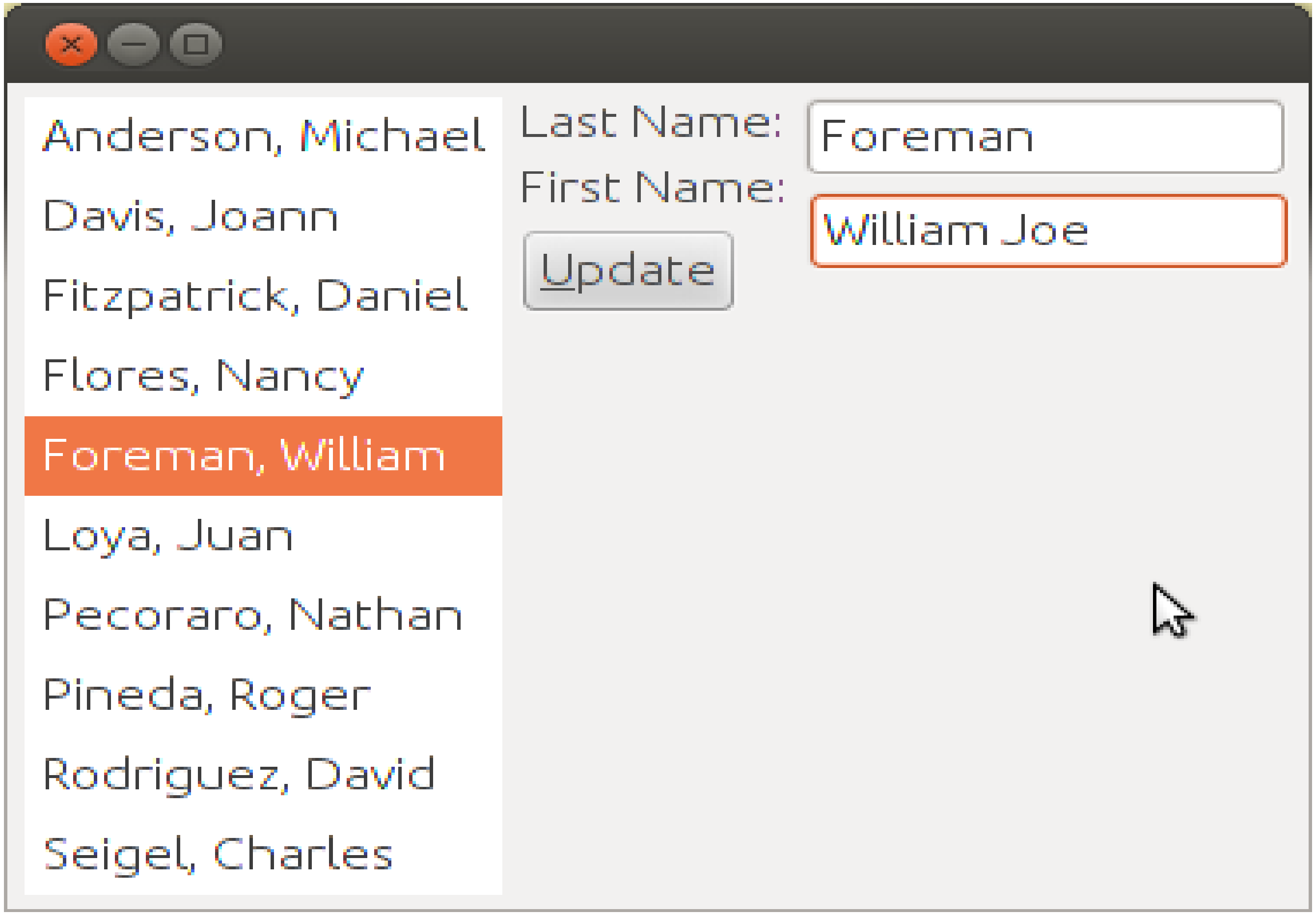
Benefits are “obvious”

Programming Languages, Libraries, and Frameworks

Prog. Languages & Libraries

- Usually many curriculum issues may constrain PL choice
- Don't overlook learning to use libraries
  - Using the fundamental libraries and frameworks that come with a language is central to “knowing a programming language”

Java Swing JList supports MVC?



Anderson, Michael
Davis, Joann
Fitzpatrick, Daniel
Flores, Nancy
Foreman, William
Loya, Juan
Pecoraro, Nathan
Pineda, Roger
Rodriguez, David
Seigel, Charles

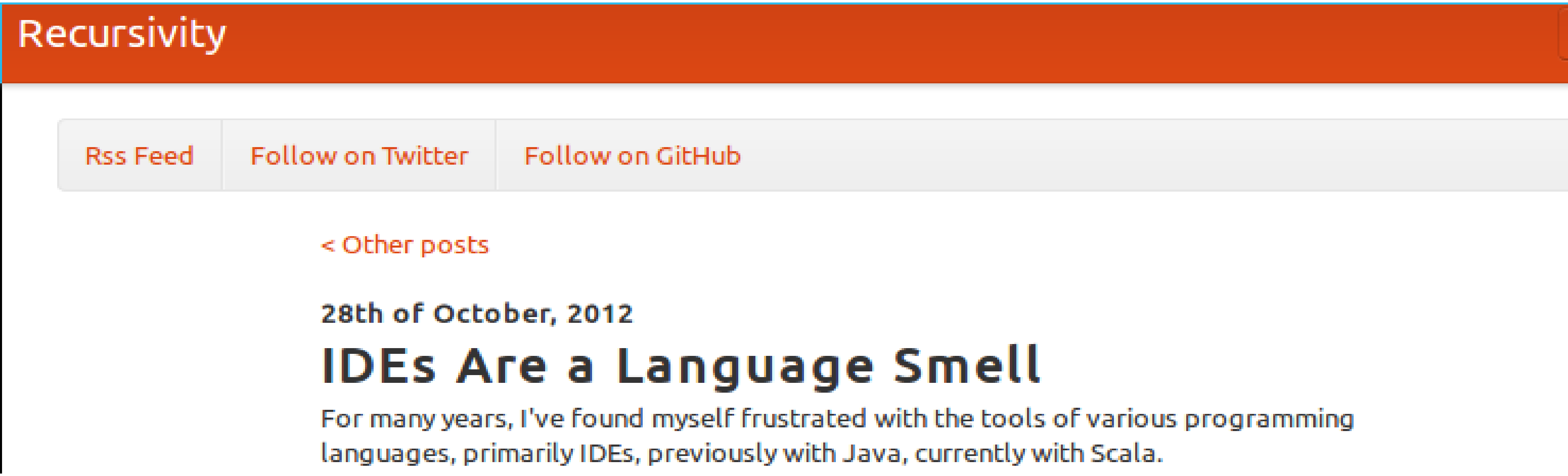
Last Name: Foreman
First Name: William Joe

Integrated Development Environments

IDEs

- Full disclosure: I hate IDEs
- “*I don’t like Eclipse because it reminds me of all the programmers I’ve worked with who can’t code if you take Eclipse away from them*” (Greg on sebastien-arbogast.com/2009/07/18/why-do-i-hate-eclipse/)
- I don't particularly even **want** to understand Eclipse!

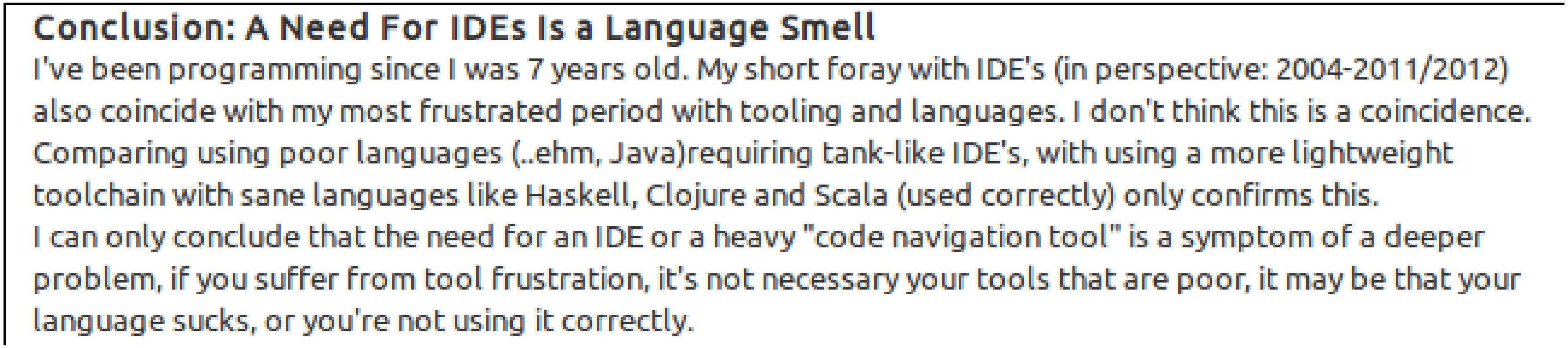
Sniff? Sniff?



28th of October, 2012

IDEs Are a Language Smell

For many years, I've found myself frustrated with the tools of various programming languages, primarily IDEs, previously with Java, currently with Scala.



Conclusion: A Need For IDEs Is a Language Smell

I've been programming since I was 7 years old. My short foray with IDE's (in perspective: 2004-2011/2012) also coincide with my most frustrated period with tooling and languages. I don't think this is a coincidence. Comparing using poor languages (..ehm, Java)requiring tank-like IDE's, with using a more lightweight toolchain with sane languages like Haskell, Clojure and Scala (used correctly) only confirms this.

I can only conclude that the need for an IDE or a heavy "code navigation tool" is a symptom of a deeper problem, if you suffer from tool frustration, it's not necessary your tools that are poor, it may be that your language sucks, or you're not using it correctly.

http://www.recursivity.com/blog/2012/10/28/ides-are-a-language-smell/

Problems with IDEs

- Overall: Students don't learn what is “behind the scenes”
- Our students are the ones who should know this! (recall: we are the experts)
- How to:
  - Organize a project
  - Build a project
  - Deploy a project

Okay, now run that from the command line.

...

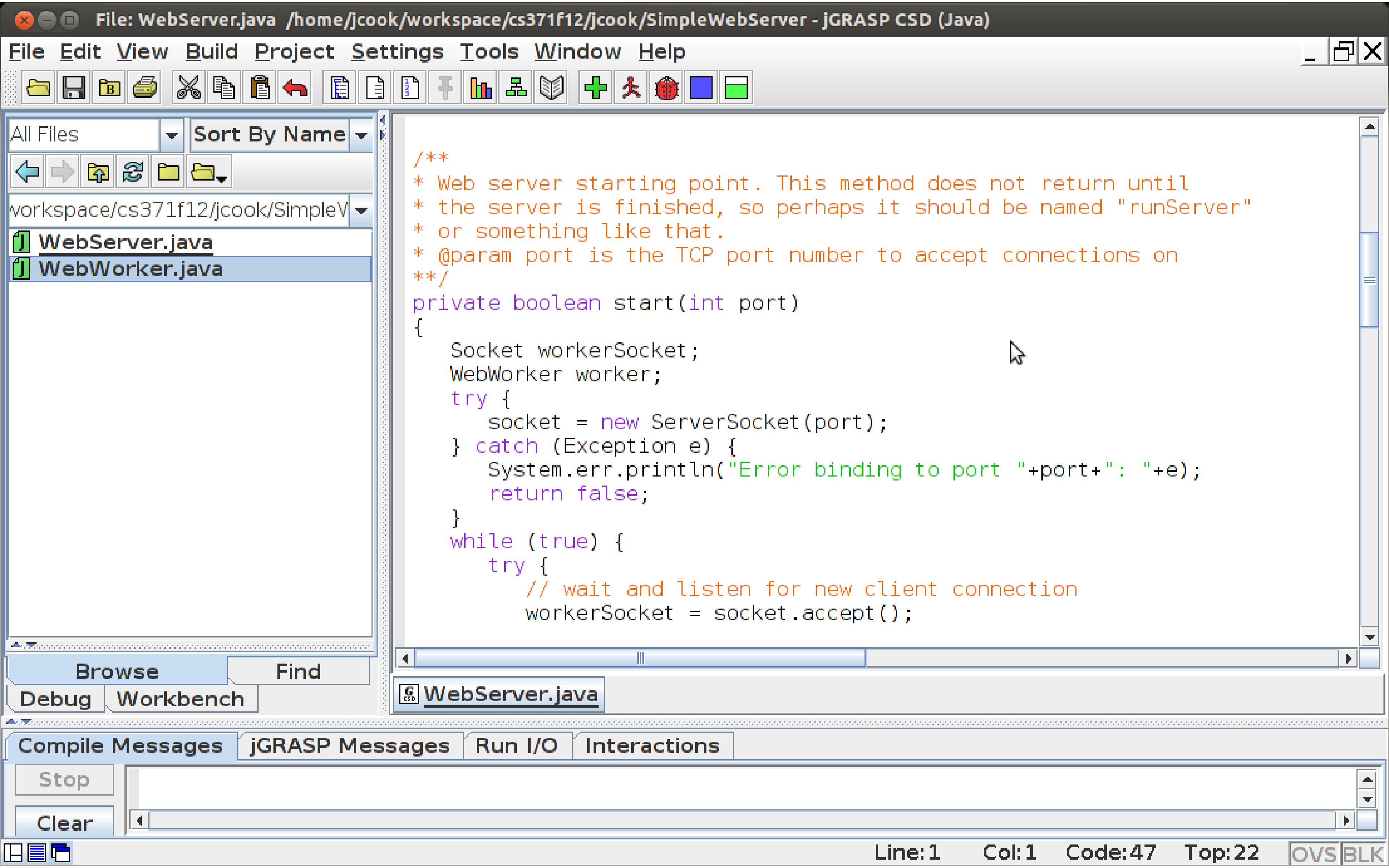
What do you mean, command line?

List of IDEs / Programming Editors

- Eclipse
- Netbeans
- Anjunta http://www.anjuta.org/
- Visual Studio (MS)
- Xcode (Mac)
- IntelliJ (commercial)
- jGrasp
- jEdit, Kate, Gedit, Vim

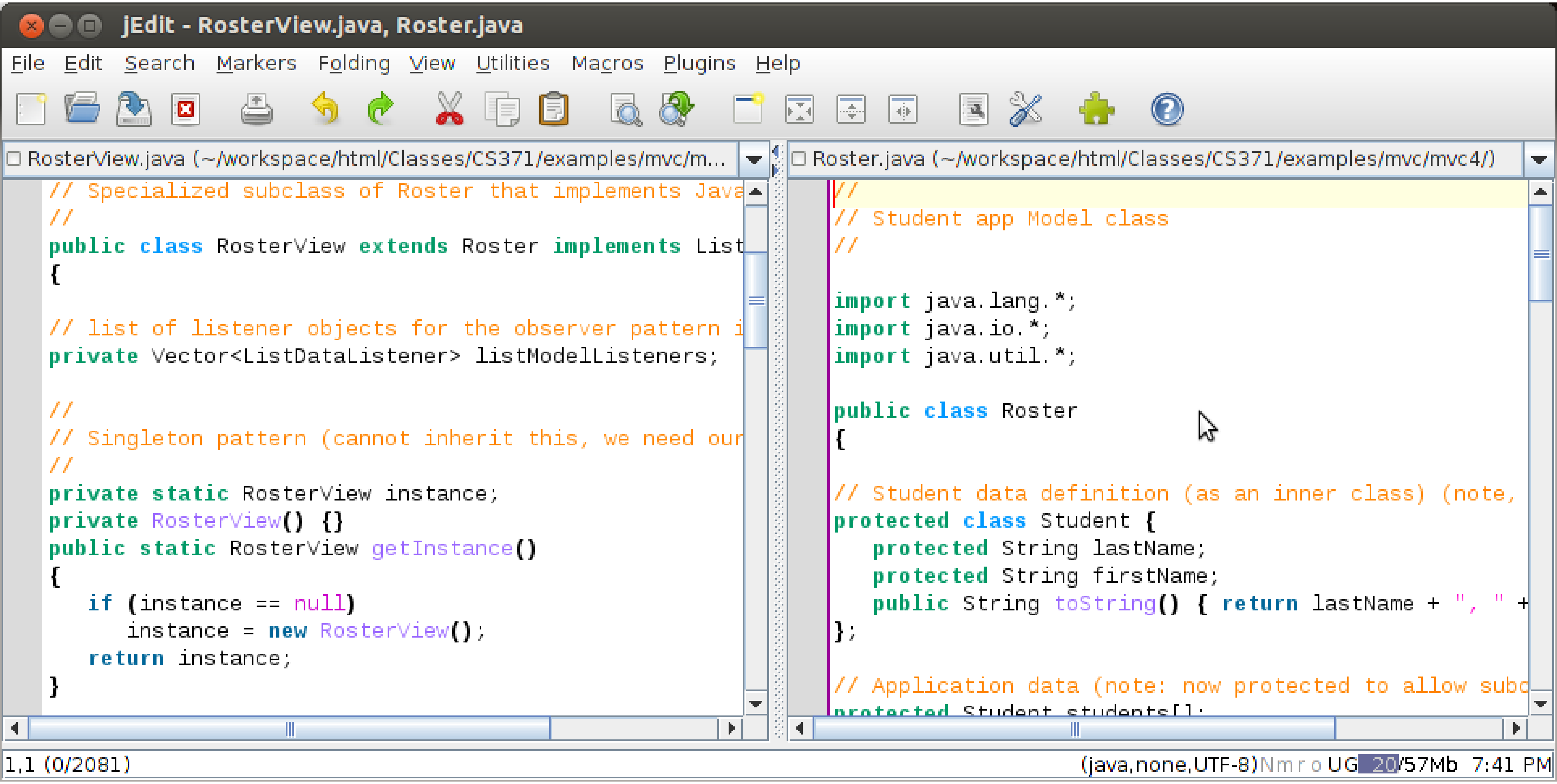
jGRASP

- Free lightweight IDE
- A university project! (Auburn U)
- Mostly Java, but other languages
- About right for student assignments and projects


File: WebServer.java /home/jcook/workspace/cs371f12/jcook/SimpleWebServer - jGRASP CSD (Java)
File Edit View Build Project Settings Tools Window Help
All Files
Sort By Name
workspace/cs371f12/jcook/SimpleV
WebServer.java
WebWorker.java
Browse
Find
Debug
Workbench
WebServer.java
Compile Messages
jGRASP Messages
Run I/O
Interactions
Stop
Clear
Line:1
Col:1
Code:47
Top:22
OVS
BLK

```
/**
* Web server starting point. This method does not return until
* the server is finished, so perhaps it should be named "runServer"
* or something like that.
* @param port is the TCP port number to accept connections on
**/
private boolean start(int port)
{
   Socket workerSocket;
   WebWorker worker;
   try {
      socket = new ServerSocket(port);
   } catch (Exception e) {
      System.err.println("Error binding to port "+port+": "+e);
      return false;
   }
   while (true) {
      try {
         // wait and listen for new client connection
         workerSocket = socket.accept();
```

jEdit



□ RosterView.java (~/workspace/html/Classes/CS371/examples/mvc/m...

```
// Specialized subclass of Roster that implements Java
//
public class RosterView extends Roster implements List
{

// list of listener objects for the observer pattern i
private Vector<ListDataListener> listModelListeners;

//
// Singleton pattern (cannot inherit this, we need our
//
private static RosterView instance;
private RosterView() {}
public static RosterView getInstance()
{
   if (instance == null)
      instance = new RosterView();
   return instance;
}
```

□ Roster.java (~/workspace/html/Classes/CS371/examples/mvc/mvc4/)

```
//
// Student app Model class
//

import java.lang.*;
import java.io.*;
import java.util.*;

public class Roster
{

// Student data definition (as an inner class) (note,
protected class Student {
   protected String lastName;
   protected String firstName;
   public String toString() { return lastName + ", " +
};

// Application data (note: now protected to allow subc
protected Student students[];
```

1,1 (0/2081) (java,none,UTF-8)Nmro UG 20/57Mb 7:41 PM

Project Repositories
(Revision Control Systems)

Revision Control Systems

- Usual question:
  - Subversion,
  - Git, or
  - Mercurial?
- But why not just use the web?
  - Many free project-hosting sites nowadays
- We'll talk about both

Question: Is more freedom always better?

I prefer Subversion over Git
precisely because
it is more constrained

Subversion vs. Git (bazaar, ...)

- Collaboration:
  - Subversion: central repository, checkin/out
  - Git: just clone the repo!
- Managing deviations
  - Subversion: branches take work
  - Git: Branch all the time!
- Workflow
  - Subversion: update/commit centrally
  - Git: any way you want! push/pull whenever!

How I use Subversion (1)

- I run an svn server (svnserve)
  - svnserve -d -r /jcd1/servers/svn
  - One server can serve multiple repositories
  - No direct file access (nor svn+ssh)
- I create a new repository for each class
  - And I have ones for grad student projects, papers, research projects, web work, …
  - svnadmin create /jcd1/servers/svn/cs371f2012

How I use Subversion (2)

- In repo/conf/passwd, create student IDs and passwords
  - Same as user account ID, password is initials plus digits from their student ID
- In repo/conf/svnserve.conf, set
  - anon-access = none   (disable anonymous read)
  - auth-access = write   (allow authorized to write)
  - password-db = passwd   (point to password file)
  - authz-db = authz   (point to authorization file)
  - realm = Classes   (allows me easy access to all)

How I use Subversion (3)

- Checkout empty repo, then add a top-level directory for every student
  - Shell scripting makes this easy!
  - Name the directory same as the account id
- In repo/conf/authz, for each student add

```
[/studentid]
studentid = rw
jcook = rw
taID = rw
* =
```

How I use Subversion (4)

- Done – now I have individual student repositories not readable by other students
- Assignments are submitted by student committing an assignmentN directory
  - TA puts grade file in directory when graded
- Teams:
  - Create teamN top-level directory for each team, give all team members (and me and TA) permission in repo/conf/authz

How I use Subversion (5)

- Benefits:
  - Students must use it; have basic commands down by the time they do teamwork
- Drawbacks:
  - Students initially think “use repository to submit assignment” and this can ossify
  - Students do not have to embrace “coordination using a repository” if they don't want to
  - But many teams effectively embrace team repository

“Whole Project” Systems

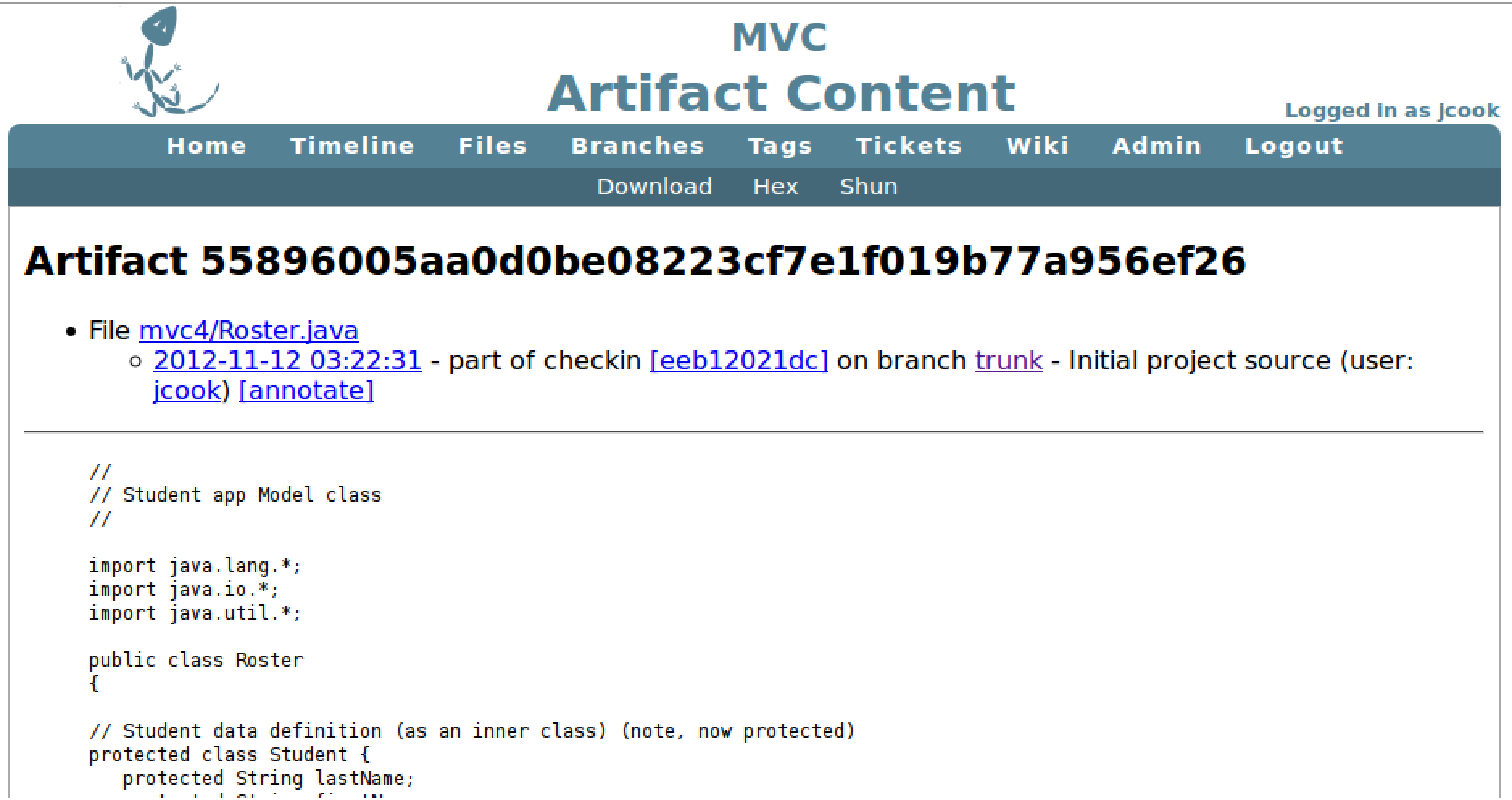
- Initial online “project hosting” sites were mostly repository plus some web links
  - Sourceforge
- Online repositories quickly transitioned to “whole project” support
  - Repository, Bug/issue tracker, documentation, Wiki
- And many more are available
  - Google code
  - Github, Gitorious
  - Gnu Bazaar http://bazaar.canonical.com/en/
  - JavaForge
  - Launchpad (Canonical)

Hosting Site Issues

- Main issue for me: privacy and ownership
- What is the EULA for the site?
- What is required of the students to use it?
    - E.g.: Google Code requires Google id
- Can you legally require students to use such a site?
    - Discussion?

Whole Project Local Systems

- Trac (http://trac.edgewall.org)
  - Documentation, issue tracker, project wiki
  - Integrates with repository (subversion, git)
  - May try it out this summer (Christmas?)
- Fossil (http://www.fossil-scm.org)
  - Integrated repository system
  - (looks very interesting!)



MVC

Artifact Content

Logged in as jcook

Artifact 55896005aa0d0be08223cf7e1f019b77a956ef26

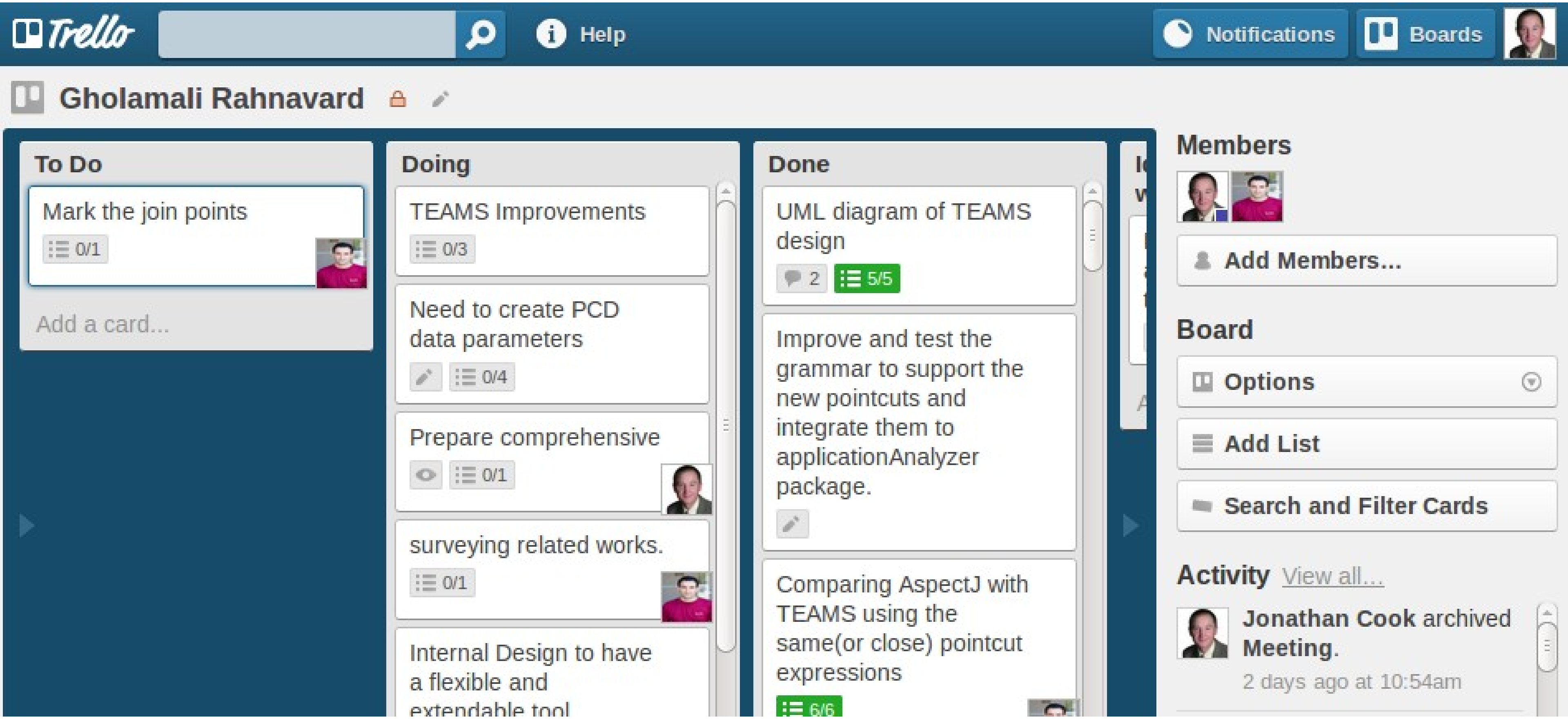
- File mvc4/Roster.java
  - 2012-11-12 03:22:31 - part of checkin [eeb12021dc] on branch trunk - Initial project source (user: jcook) [annotate]

```
//
// Student app Model class
//

import java.lang.*;
import java.io.*;
import java.util.*;

public class Roster
{

// Student data definition (as an inner class) (note, now protected)
protected class Student {
   protected String lastName;
```

Standalone Issue/project tracker

- Bugzilla and others do not meet my requirements
  - Not easy to install, uses at least mysql
- Roundup is a simple issue tracker
  - Easy to set up (user setup?)
  - 1 server, multiple trackers (1 per team)
- Your LMS may have project team support
  - Discussions, wiki, other
- Trello.com is nice! (must trust EULA)

Trello (Fog Creek Software)



Gholamali Rahnavard

To Do

Mark the join points
0/1

Add a card...

Doing

TEAMS Improvements
0/3

Need to create PCD data parameters
0/4

Prepare comprehensive
0/1

surveying related works.
0/1

Internal Design to have a flexible and extendable tool

Done

UML diagram of TEAMS design
2 5/5

Improve and test the grammar to support the new pointcuts and integrate them to applicationAnalyzer package.

Comparing AspectJ with TEAMS using the same(or close) pointcut expressions
6/6

Members

Add Members...

Board

Options

Add List

Search and Filter Cards

Activity View all...

Jonathan Cook archived Meeting.
2 days ago at 10:54am

Build Tools

Build Tools

- Make and ant are standard
- Always will be others...
  - Rake,cake,maven,boost.build, jam
- Then meta-build tools
  - Autoconf/automake, Cmake, …
- We expose students to make and ant
- Discussion?

Code Documentation Tools

Code Documentation Tools

- Javadoc and doxygen
- Danger: tools “work” without students needing any true doc-able comments
- So must establish expectations as to what the students must document
  - Methods, all parameters, return value
  - Generate docs for private members too

Testing Tools

Unit Testing Tools

- I stick with Junit
  - Easy to install and use for the basic capability
- Students can install and use directly
- Discussion: Anyone doing something cool with unit testing?

Coverage Tools

- C/C++/multi-language:
  - C/C++: gcov (Gnu), lcov (Linux front end to gcov), covtool (sourceforge, source instr, last 4/2010), trucov (google code, Jul 2010), xcover (looks dead), SquishCoco (froglogic, was TestCocoon, free for non-comm?)
- Two reasonable Java OSS tools:
  - Cobertura (6/2011)
  - EMMA (6/2011)
- I like EMMA
  - One jar, students can download and use
  - Easy to use

Emma: easy to use

```
java -cp ~/bin/emma.jar emmarun -cp . IfCounter basictest.txt
```

- does everything, including instrumentation & text report

```
java -cp ~/bin/emma.jar emmarun -r html -sp . -cp . IfCounter
…
```

- HTML report output, with highlighted source

```
java -cp ~/bin/emma.jar emma instr [options] [classes/jar]
```

- offline instrumentation

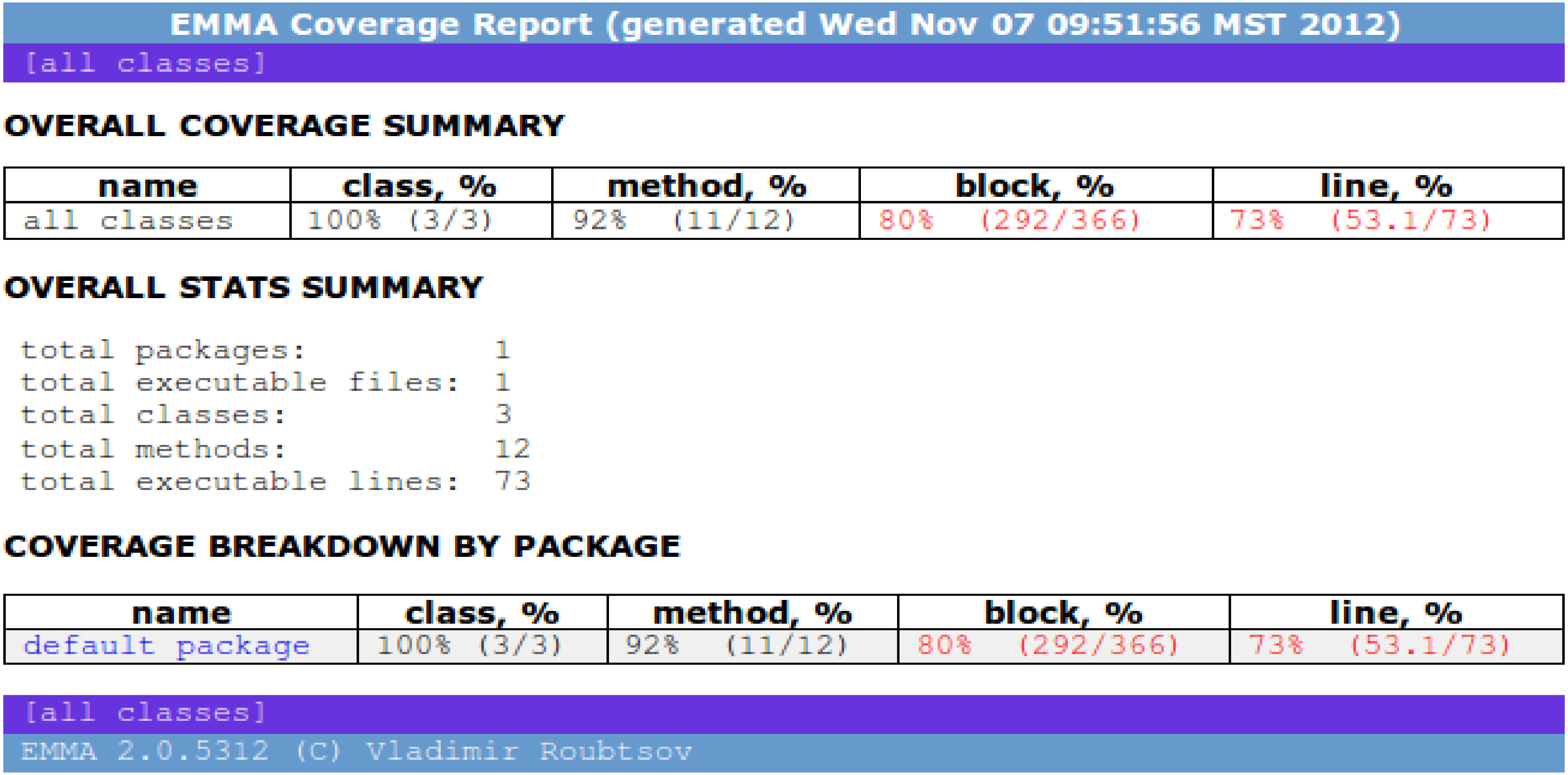
```
java -cp [include emma.jar] instrumented-prog
```

- run java program that is already instrumented

```
java -cp ~/bin/emma.jar emma report [options]
```

- generate a report from instrumentation data

Emma Output 1



EMMA Coverage Report (generated Wed Nov 07 09:51:56 MST 2012)

[all classes]

OVERALL COVERAGE SUMMARY

name	class, %	method, %	block, %	line, %
all classes	100% (3/3)	92% (11/12)	80% (292/366)	73% (53.1/73)

OVERALL STATS SUMMARY

```
total packages:          1
total executable files:  1
total classes:           3
total methods:           12
total executable lines:  73
```

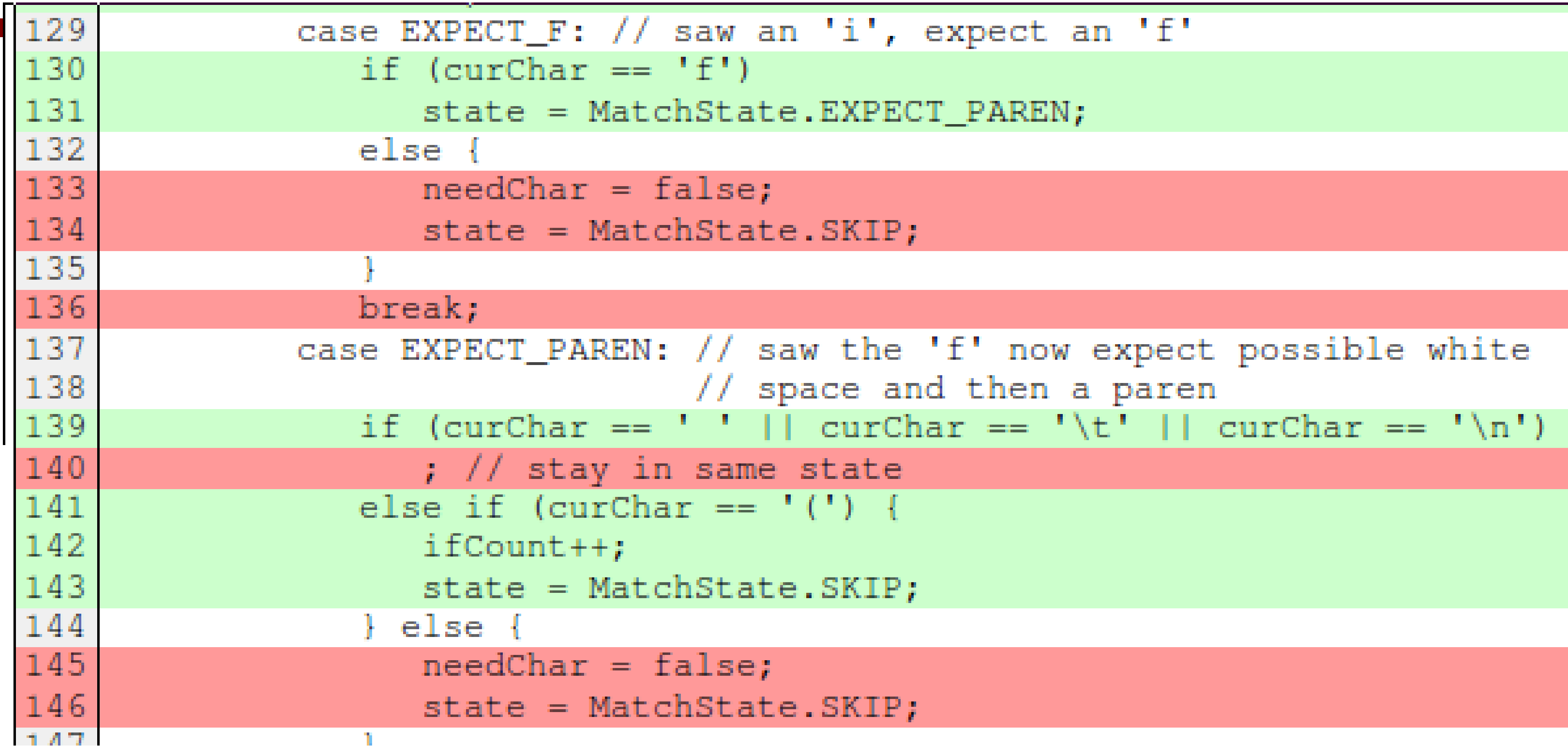
COVERAGE BREAKDOWN BY PACKAGE

name	class, %	method, %	block, %	line, %
default package	100% (3/3)	92% (11/12)	80% (292/366)	73% (53.1/73)

[all classes]

EMMA 2.0.5312 (C) Vladimir Roubtsov

Emma Output 2



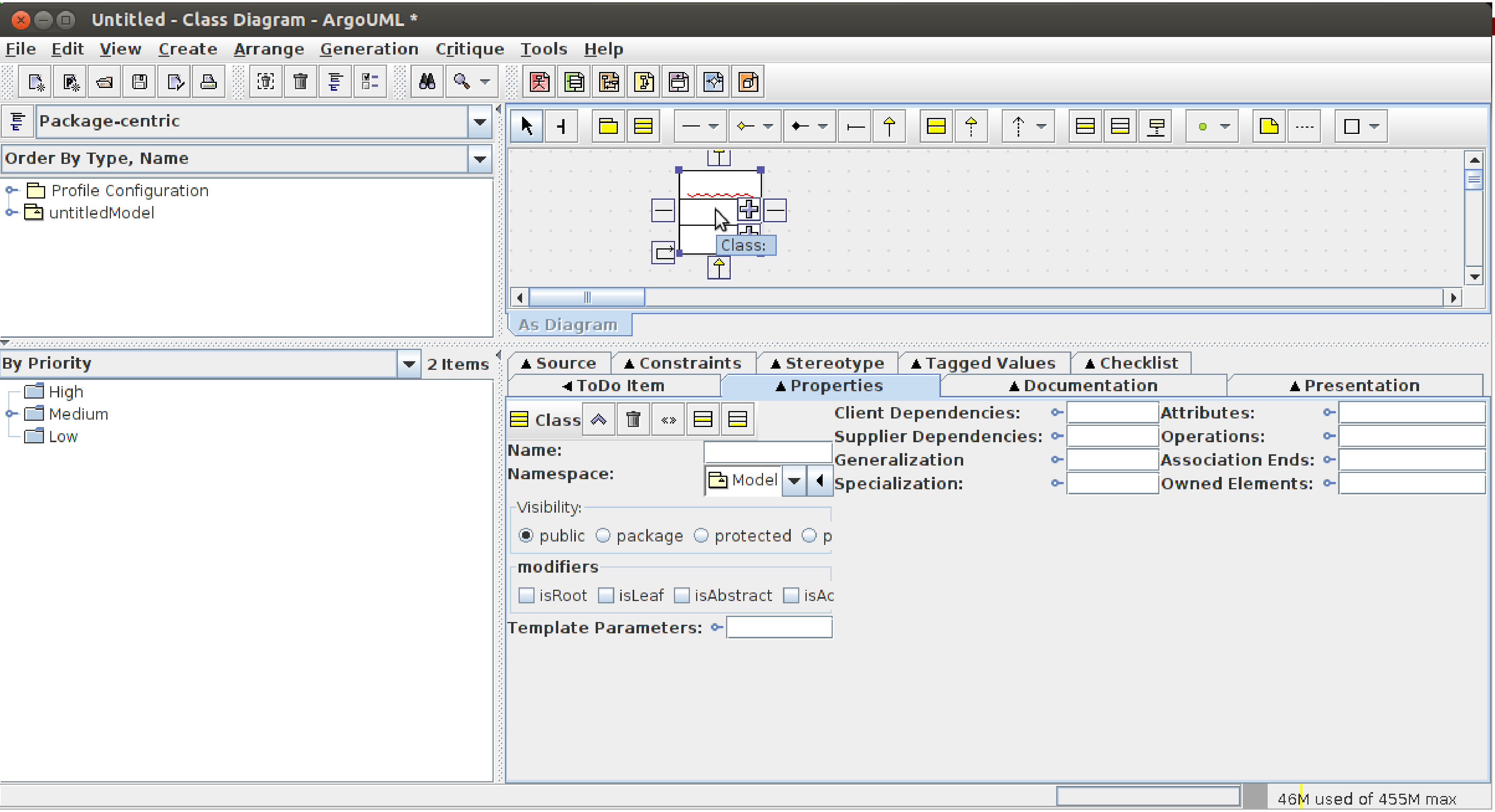
```
case EXPECT_F: // saw an 'i', expect an 'f'
   if (curChar == 'f')
      state = MatchState.EXPECT_PAREN;
   else {
      needChar = false;
      state = MatchState.SKIP;
   }
   break;
case EXPECT_PAREN: // saw the 'f' now expect possible white
                   // space and then a paren
   if (curChar == ' ' || curChar == '\t' || curChar == '\n')
      ; // stay in same state
   else if (curChar == '(') {
      ifCount++;
      state = MatchState.SKIP;
   } else {
      needChar = false;
      state = MatchState.SKIP;
   }
```

Diagramming Tools

UML Tools

- ArgoUML is a well-known UML tool
  - See next screenshot
- I like Umlet
  - Simple, easy to install, use
  - Probably not for a commercial project, but...

Any Questions?



Name:

Namespace: Model

Visibility: ● public ○ package ○ protected ○ p

modifiers: □ isRoot □ isLeaf □ isAbstract □ isAc

Template Parameters:

Client Dependencies:

Supplier Dependencies:

Generalization:

Specialization:

Attributes:

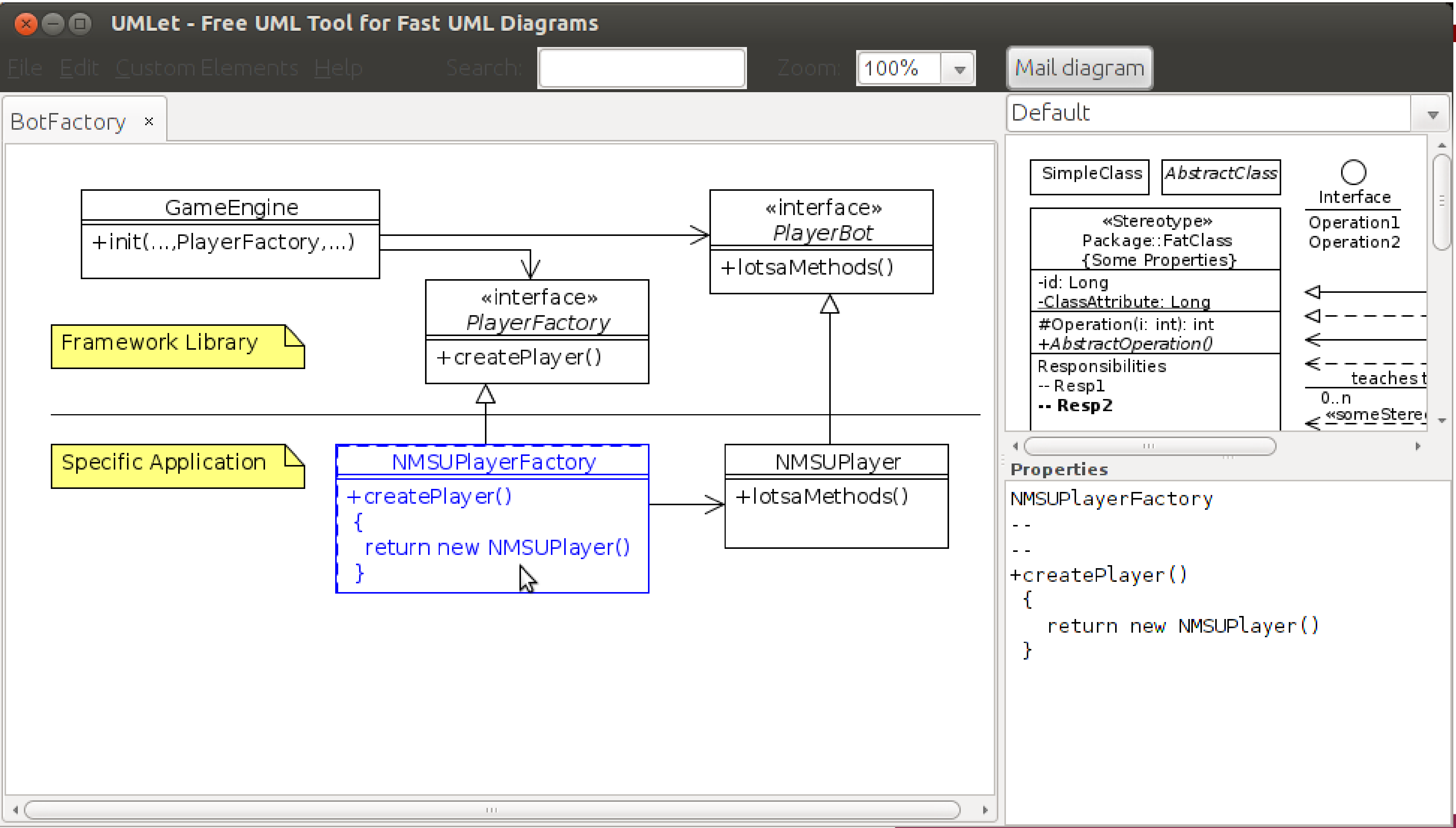
Operations:

Association Ends:

Owned Elements:

46M used of 455M max

Aaahhh...

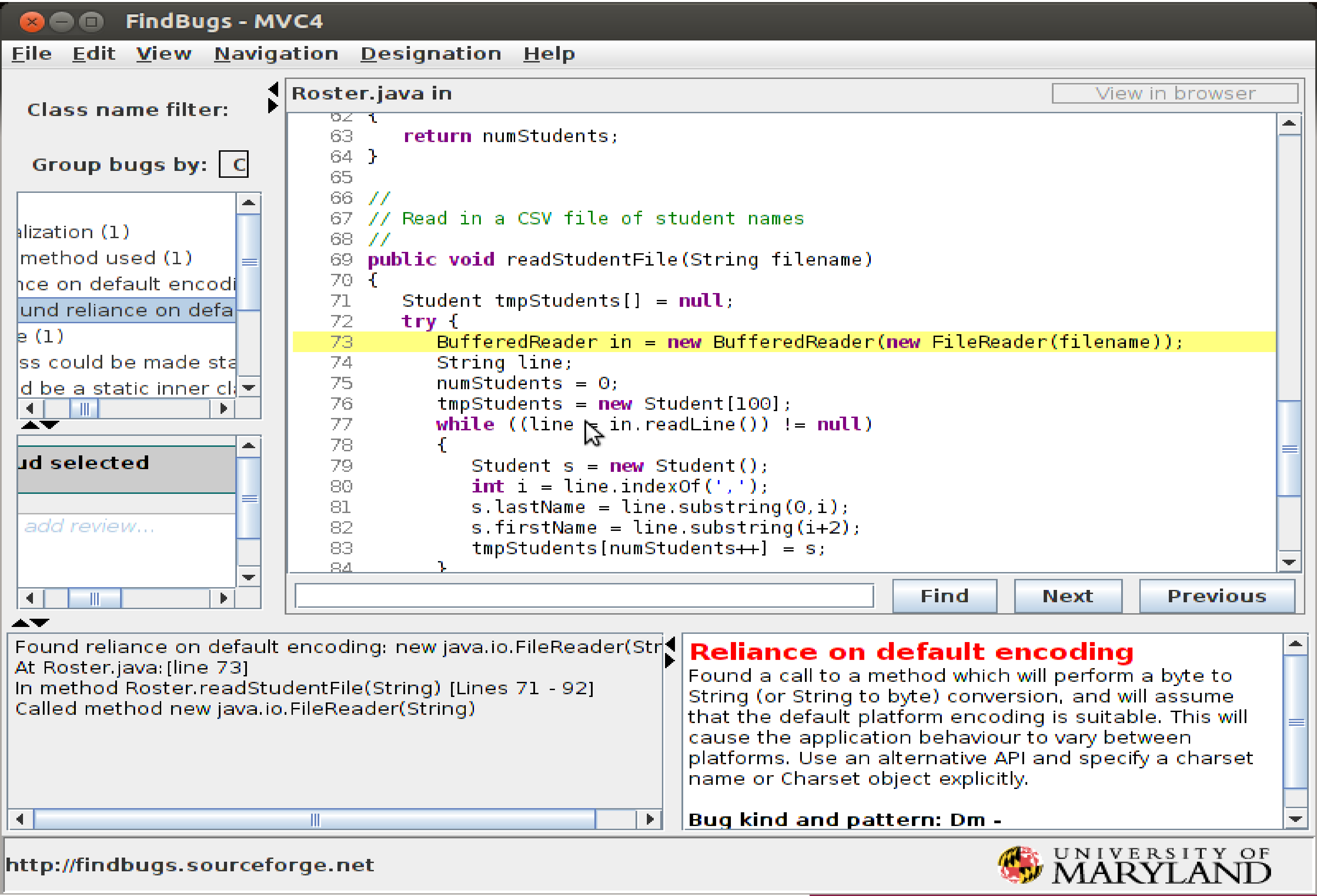


```
NMSUPlayerFactory
--
--
+createPlayer()
 {
   return new NMSUPlayer()
 }
```

Verification Tools

Verification Tools

- Honestly, I don't do much here at the undergraduate level
- Graduate, I walk through
  - JML, ESCJava, Spin as different classes of analysis (dynamic, incomplete static, sound and complete static)
- FindBugs is one tool I sometimes introduce to undergrads (if time permits)
  - Easy to install, use, benefits are “obvious”



ılization (1)
method used (1)
ıce on default encodi
und reliance on defa
e (1)
ss could be made sta
d be a static inner cl

ıd selected

```
62 {
63     return numStudents;
64 }
65
66 //
67 // Read in a CSV file of student names
68 //
69 public void readStudentFile(String filename)
70 {
71     Student tmpStudents[] = null;
72     try {
73         BufferedReader in = new BufferedReader(new FileReader(filename));
74         String line;
75         numStudents = 0;
76         tmpStudents = new Student[100];
77         while ((line = in.readLine()) != null)
78         {
79             Student s = new Student();
80             int i = line.indexOf(',');
81             s.lastName = line.substring(0,i);
82             s.firstName = line.substring(i+2);
83             tmpStudents[numStudents++] = s;
84         }
```

Found reliance on default encoding: new java.io.FileReader(Str
At Roster.java:[line 73]
In method Roster.readStudentFile(String) [Lines 71 - 92]
Called method new java.io.FileReader(String)

Reliance on default encoding

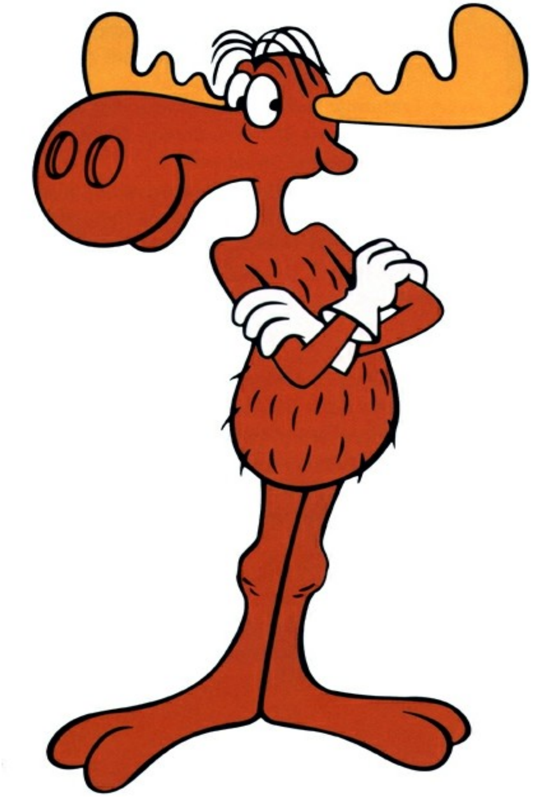
Found a call to a method which will perform a byte to String (or String to byte) conversion, and will assume that the default platform encoding is suitable. This will cause the application behaviour to vary between platforms. Use an alternative API and specify a charset name or Charset object explicitly.

Bug kind and pattern: Dm -

LMS Tools

What does your LMS do?

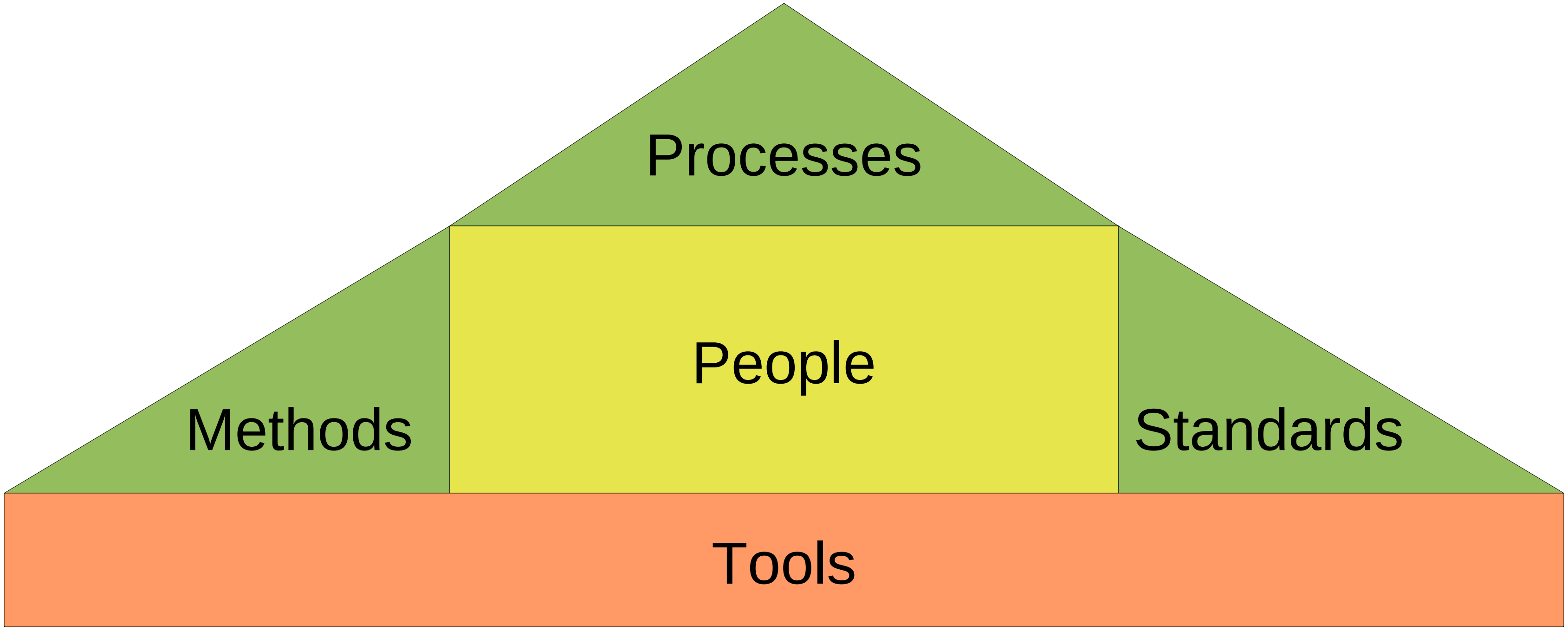
- NMSU switched from Blackboard to Canvas (instructure.com)
- Can create student groups (teams)
- Can create discussion topics
  - e.g., one per team, but probably public
- Groups have an activity message board
- Plugins allow collaborative documents



prosportstickers.com

“Any questions?”

Software Engineering



Processes

Methods

People

Standards

Tools