How to Narrow Down What to Test

by
Zsolt Fabok
2013-06-05

@ZsoltFabok

or

#xp2013

"I get paid for code that works, not for tests, so my philosophy is to test as little as possible to reach a given level of confidence (I suspect this level of confidence is high compared to industry standards, but that could just be hubris). If I don't typically make a kind of mistake (like setting the wrong variables in a constructor), I don't test for it. I do tend to make sense of test errors, so I'm extra careful when I have logic with complicated conditionals. When coding on a team, I modify my strategy to carefully test code that we, collectively, tend to get wrong."

Kent Beck - September 10, 2010

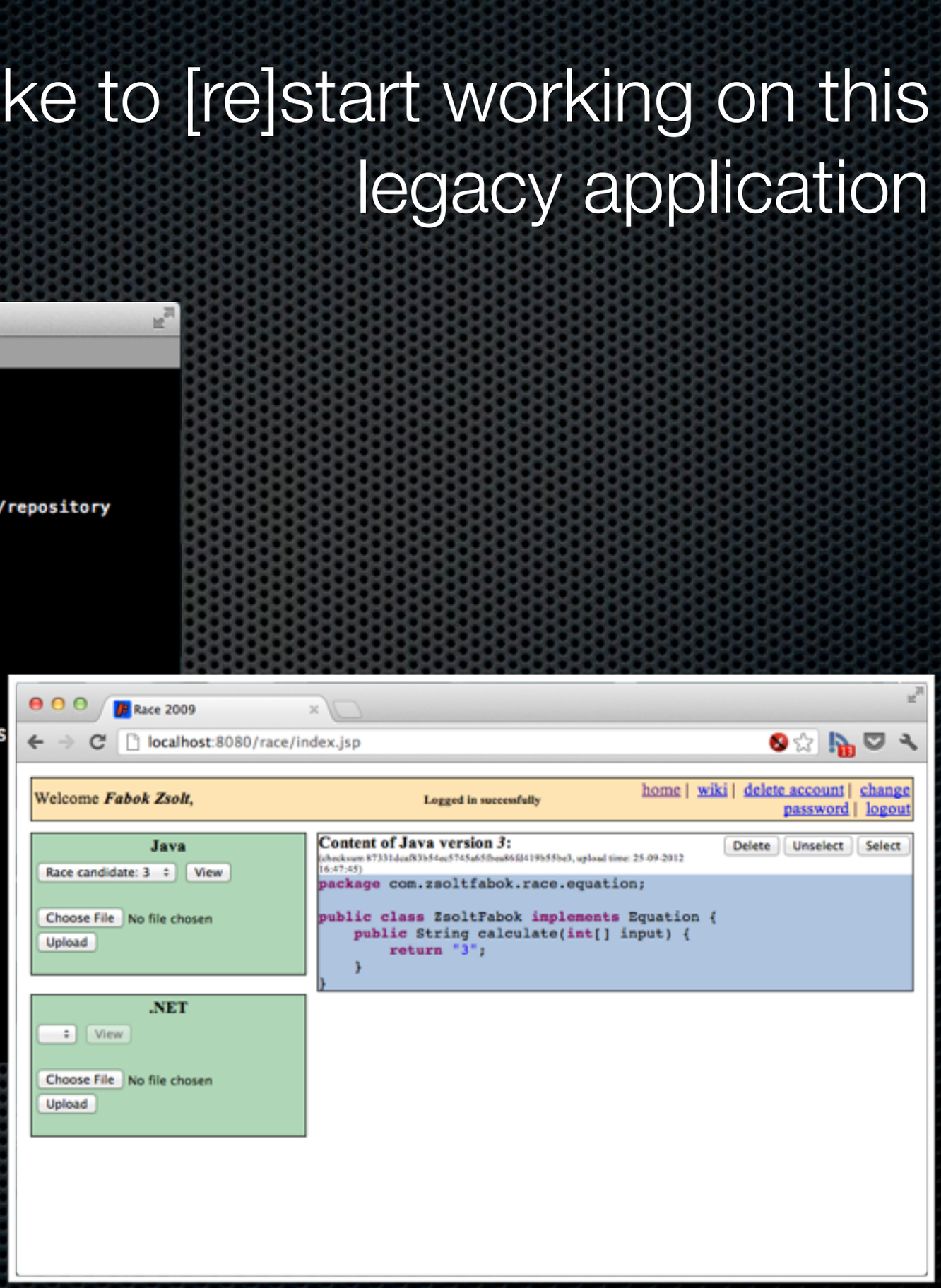
quote: http://stackoverflow.com/questions/153234/how-deep-are-your-unit-tests/153565#153565

I’d like to [re]start working on this legacy application

Terminal — zsh — 91×33

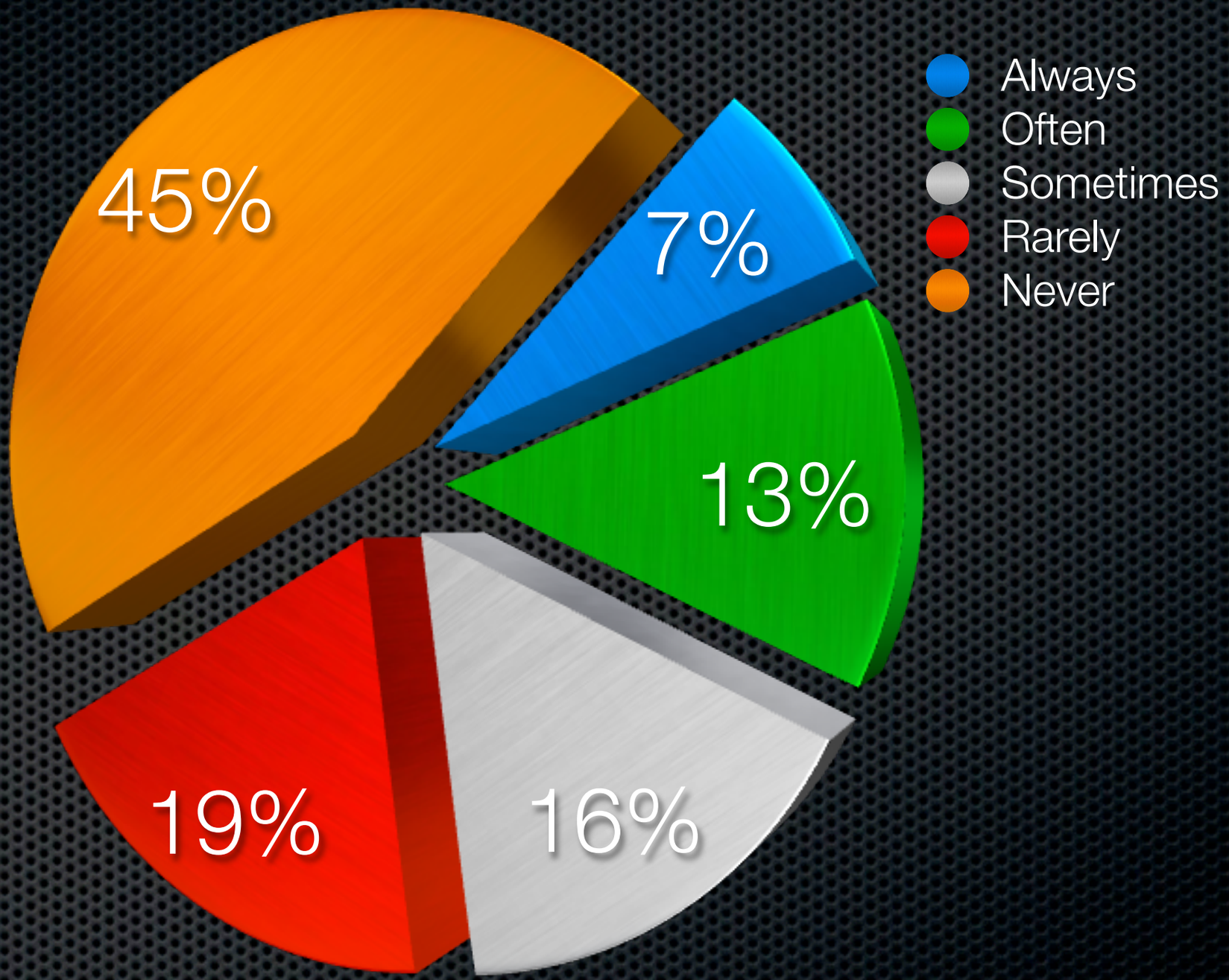
zsh

```
~//Race2009 % ant verifier
Buildfile: /Users/zsolt/Race2009/build.xml

clean.java:

verifier:
[harvester] Collect candidates from user repository: /Users/zsolt/Race2009/repository
[harvester] zsolt.fabok.ext      JAVA    Check: NONE  Copy: NONE
[harvester] zsolt.fabok.ext      NET     Check: OK    Copy: OK
[harvester] zsolt.fabok          JAVA    Check: OK    Copy: OK
[harvester] zsolt.fabok          NET     Check: OK    Copy: NONE
[harvester] Done

build.competitor.java:

 [verifier] Verifying competitors:
 [verifier] zsolt.fabok.ext      NET     Verified: YES Undo: NO   Mail: NO
 [verifier] zsolt.fabok          JAVA    Verified: NO  Undo: N/A  Mail: YES
 [verifier] Done

BUILD SUCCESSFUL
Total time: 1 seconds
~/Code/java/evorace/cruisecontrol/projects/EvoRace2009_Verifier %
```



My “where to start” list

#1 Determine which parts of the code are really used



The good old Standish Group Study

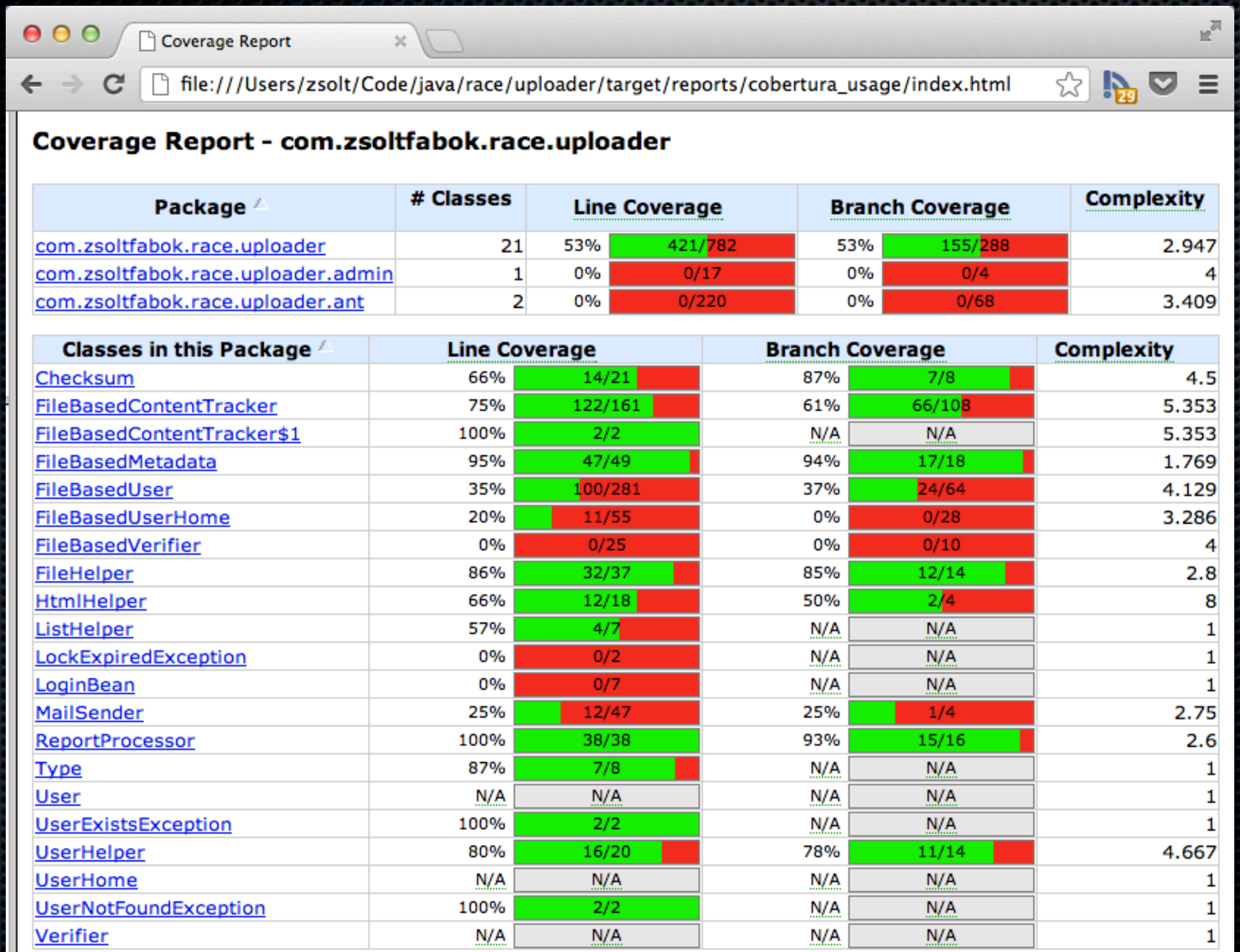
The goal is to find those features which are always or often used.

By studying coverage, access logs, traces, web analytics, heat maps, etc.

Let’s have a look at the coverage (using instrumented class files):

```
% cp jetty/cobertura.ser web.ser
% cp uploader/cobertura.ser ant.ser
% ant usage_coverage

usage_coverage:
[cobertura-merge] Cobertura: Loaded information on 12 classes.
[cobertura-merge] Cobertura: Loaded information on 11 classes.
[cobertura-merge] Cobertura: Saved information on 16 classes.
[cobertura-report] Cobertura: Loaded information on 16 classes.
[cobertura-report] Report time: 600ms

BUILD SUCCESSFUL
Total time: 2 seconds
```

Example #1: overview



Package	# Classes	Line Coverage	Branch Coverage	Complexity
com.zsoltfabok.race.uploader	21	53% 421/782	53% 155/288	2.947
com.zsoltfabok.race.uploader.admin	1	0% 0/17	0% 0/4	4
com.zsoltfabok.race.uploader.ant	2	0% 0/220	0% 0/68	3.409

Classes in this Package	Line Coverage	Branch Coverage	Complexity
Checksum	66% 14/21	87% 7/8	4.5
FileBasedContentTracker	75% 122/161	61% 66/108	5.353
FileBasedContentTracker$1	100% 2/2	N/A N/A	5.353
FileBasedMetadata	95% 47/49	94% 17/18	1.769
FileBasedUser	35% 100/281	37% 24/64	4.129
FileBasedUserHome	20% 11/55	0% 0/28	3.286
FileBasedVerifier	0% 0/25	0% 0/10	4
FileHelper	86% 32/37	85% 12/14	2.8
HtmlHelper	66% 12/18	50% 2/4	8
ListHelper	57% 4/7	N/A N/A	1
LockExpiredException	0% 0/2	N/A N/A	1
LoginBean	0% 0/7	N/A N/A	1
MailSender	25% 12/47	25% 1/4	2.75
ReportProcessor	100% 38/38	93% 15/16	2.6
Type	87% 7/8	N/A N/A	1
User	N/A N/A	N/A N/A	1
UserExistsException	100% 2/2	N/A N/A	1
UserHelper	80% 16/20	78% 11/14	4.667
UserHome	N/A N/A	N/A N/A	1
UserNotFoundException	100% 2/2	N/A N/A	1
Verifier	N/A N/A	N/A N/A	1

Example #2: execution

Coverage Report

file:///Users/zsolt/Code/java/race/uploader/target/reports/cobertura_usage/index.html

```
 89 393            if (!disqualifiedReached) {
 90                     // Evaluate competitors
 91 381                 if (line.contains(NOT_QUALIFIED_LINE)) {
 92   3                     disqualifiedReached = true;
 93                     } else {
 94                         // Get competitors
 95 378                  Matcher matcher = competitorPattern.matcher(line);
 96 378                  matcher.find();
 97 378                  if (matcher.matches()) {
 98 180                      competitors.add(matcher.group(1));
 99                      }
100 378                 }
101                 } else {
102                     // Get disqualified competitors
103  12                 Matcher matcher = disqualifiedPattern.matcher(line);
104  12                 matcher.find();
105  12                 if (matcher.matches()) {
106                         // Found another disqualified competitor
107   3                     if (disqualifiedCompetitor != null) {
108                             // Save the previous competitor before starting
109                             // with the new one
110                             disqualifiedCompetitors.put(disqualifiedCompetitor,
111                                     disqualifiedReason.toString());
112                             disqualifiedCompetitor = null;
113                             disqualifiedReason = new StringBuilder();
114                         }
115   3                     disqualifiedCompetitor = matcher.group(1);
116   3                     disqualifiedReason.append(matcher.group(2) + "\n");
117                     } else {
118                         // Not a "main" line for a competitor, if
119                         // the competitor is set, it is sure that the
120                         // the lines belong to the reason
121   9                     if (disqualifiedCompetitor != null) {
122   9                         disqualifiedReason.append(line + "\n");
123                         }
124                     }
125  12             }
126         }
```

not even executed

Example #3: number of execution

Coverage Report

file:///Users/zsolt/Code/java/race/uploader/target/reports/cobertura_usage/index.html

```
179          // Recreated history
180     3    history.load();
181     3    history.removeAll();
182     3    for(String userName : userHome.listUsers()) {
183              try {
184   111            String resultFsckMessage = MESSAGE_NONE;
185   111            String resultHarvestMessage = MESSAGE_NONE;
186   111            User user = userHome.lockUser(userName);
187                  // Verify user home, copy candidates and update
188                  // candidates list for both versions
189   111            if (!user.getVersions(Type.JAVA).isEmpty()) {
190    42                if (user.integrityCheck(Type.JAVA)) {
191    42                    resultFsckMessage = MESSAGE_OK;
192                          try {
193    42                        if (harvestJava(user)) {
194    36                            resultHarvestMessage = MESSAGE_OK;
195                              } else {
196     6                        resultHarvestMessage = MESSAGE_NONE;
197                              }
198                          } catch (IOException e) {
199                            resultHarvestMessage = MESSAGE_FAILED;
200                            log(userName + " JAVA IO error: " + e.getMessage());
201    42                    }
202                      } else {
203                          resultFsckMessage = MESSAGE_FAILED;
204                      }
205                  } else {
206    69                resultHarvestMessage = MESSAGE_NONE;
207                  }
208   111            String message = String.format(MESSAGE_PATTERN,
209                          userName, Type.JAVA.toString(),
210                          resultFsckMessage, resultHarvestMessage);
211   111            log(message);
212
213   111            if (!user.getVersions(Type.NET).isEmpty()) {
214    66                if (user.integrityCheck(Type.NET)) {
215    66                    resultFsckMessage = MESSAGE_OK;
216                          try {
217    66                        if (harvestNet(user)) {
```

.NET wins

FileBasedMetadata (usage)

FileHelper (usage)

#2 Find out which parts of the code change often

By checking how many times a file has been committed into VCS:

```
% ./git_stat.sh

14, VerifierTask.java
13, index.jsp
11, FileBasedUserHome.java
11, FileBasedUser.java
11, FileBasedContentTracker.java
 8, IntegrityCheckTask.java
 7, MailSender.java
```

FileBasedMetadata (usage)

FileHelper (usage)

VerifierTask (changes)

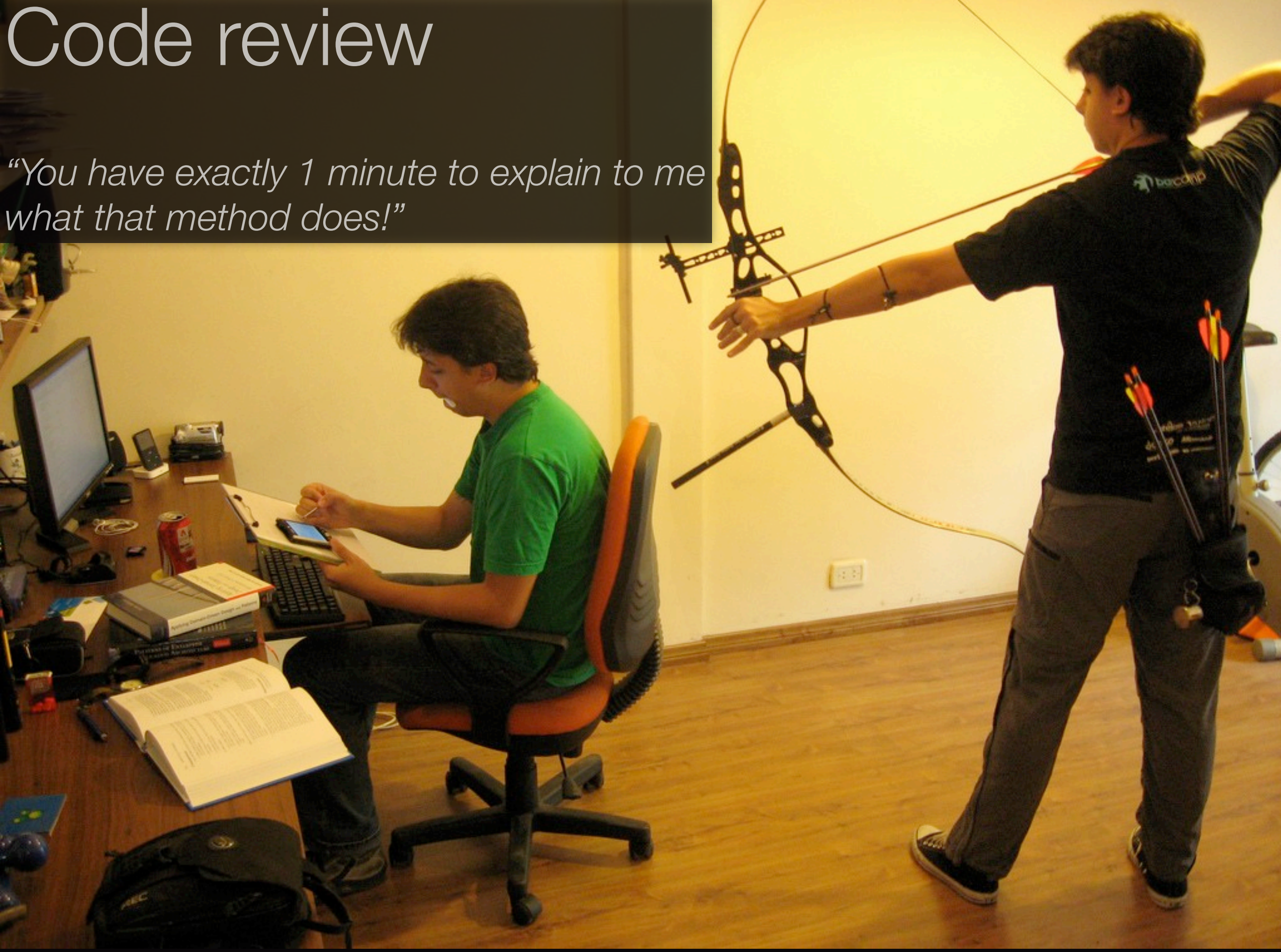
index.jsp (changes)

FileBasedUserHome (changes)

#3 Determine which part of the code changes data

Code review

"You have exactly 1 minute to explain to me what that method does!"



FileBasedMetadata (usage)

FileHelper (usage, review)

VerifierTask (changes)

index.jsp (changes)

FileBasedUserHome (changes, review)

FileBasedContentTracker (review)

Exercise 3: Code Review

#4 Determine where the code is most likely going to fail

(e.g. with static code checkers)

FindBugs (1.3.9) Analysis for

file:///Users/zsolt/Code/java/race/uploader/target/reports/findbugs.html

FindBugs (1.3.9) Analysis for

Summary | History | Browse By Categories | Browse by Packages | Info

-- All Versions -- | -- All priorities --

Stats by Bug Package

P1 P2 P3 Exp

Total number of bugs: 15

- **com.zsoltfabok.race.uploader** 8 bugs *(1/7/0/0)*
  - **com.zsoltfabok.race.uploader.FileBasedContentTracker** 2 bugs *(0/2/0/0)*
    - **Method ignores exceptional return value** 2 bugs *(0/2/0/0)*
  - **com.zsoltfabok.race.uploader.FileBasedMetadata** 1 bugs *(1/0/0/0)*
    - **Method ignores return value** 1 bugs *(1/0/0/0)*
  - **com.zsoltfabok.race.uploader.FileBasedUser** 2 bugs *(0/2/0/0)*
  - **com.zsoltfabok.race.uploader.FileHelper** 3 bugs *(0/3/0/0)*
- **com.zsoltfabok.race.uploader.ant** 7 bugs *(0/7/0/0)*
  - **com.zsoltfabok.race.uploader.ant.HarvesterTask** 4 bugs *(0/4/0/0)*
    - **Method may fail to clean up stream or resource** 2 bugs *(0/2/0/0)*
    - **Method may fail to close stream** 2 bugs *(0/2/0/0)*
  - **com.zsoltfabok.race.uploader.ant.VerifierTask** 3 bugs *(0/3/0/0)*

PMD 4.2.5 Report

file:///Users/zsolt/Code/java/race/uploader/target/reports/pmd.html

Summary

Files	Total	Priority 1	Priority 2	Priority 3	Priority 4	Priority 5
18	68	0	0	68	0	0

PMD is not helpful at the moment, but good to know about it

.Users.zsolt.Code.java.race.uploader.src.com.zsoltfabok.race.uploader.FileBasedUser

Prio	Begin Line	Method	Description
3	143	checkPassword	Avoid if (x != y) ..; else ..;
3	153	checkPassword	Avoid unnecessary if..then..else statements when returning a boolean
3	171	create	Avoid if (x != y) ..; else ..;
3	173	create	Avoid if (x != y) ..; else ..;
3	204	changePassword	Avoid if (x != y) ..; else ..;
3	260	uploadVersion	Avoid if (x != y) ..; else ..;
3	279	removeVersion	Avoid if (x != y) ..; else ..;
3	291	removeVersion	The String literal "Version is not set" appears 5 times in this file; the first occurrence is on line 291
3	347	getChecksum	Avoid if (x != y) ..; else ..;
3	364	getUploadTime	Avoid if (x != y) ..; else ..;
3	381	getContent	Avoid if (x != y) ..; else ..;
3	414	select	Avoid if (x != y) ..; else ..;
3	488		Use block level rather than method level synchronization
3	490	checkLockExpiration	These nested if statements could be combined
3	497		Use block level rather than method level synchronization
3	563		Use block level rather than method level synchronization
3	603	saveLockTimestamp	Avoid if (x != y) ..; else ..;
3	614	saveLockTimestamp	When instantiating a SimpleDateFormat object, specify a Locale
3	675	isLockTimestampExpired	When instantiating a SimpleDateFormat object, specify a Locale

CRAP Report Detail

(Sorted by Crap Load)

Overview Page | CRAP | Complexity | Coverage

Method	Complexity	Coverage	CRAP	CRAP Load
public boolean fsck() com.zsoltfabok.race.uploader . FileBasedContentTracker	17	0.00 %	306.00	17
public void execute() com.zsoltfabok.race.uploader.ant . VerifierTask	17	0.00 %	306.00	17
public void gc() com.zsoltfabok.race.uploader . FileBasedContentTracker	12	0.00 %	156.00	12
public void load(java.io.File) com.zsoltfabok.race.uploader . ReportProcessor	9	0.00 %	90.00	9
public void execute() com.zsoltfabok.race.uploader.ant . HarvesterTask	8	0.00 %	72.00	8
public synchronized void acquireLock(int, long, long) com.zsoltfabok.race.uploader . FileBasedUser	7	0.00 %	56.00	7
public boolean checkout(java.lang.String) com.zsoltfabok.race.uploader . FileBasedContentTracker	6	0.00 %	42.00	6
public boolean remove(java.lang.String) com.zsoltfabok.race.uploader . FileBasedContentTracker	6	0.00 %	42.00	6
public boolean addUser(java.lang.String)	6	0.00 %	42.00	6

FileBasedMetadata (usage)

FileHelper (usage, review, bugs)

VerifierTask (changes)

index.jsp (changes)

FileBasedUserHome (changes, review)

FileBasedContentTracker (review, bugs)

FileBasedContentTracker.fsck() (crap4j)

FileBasedContentTracker.gc() (crap4j)

HarversterTask (bugs)

VerifierTask.execute() (crap4j)

Let’s order our list and we are done!

FileHelper (usage, review, bugs)

FileBasedMetadata (usage)

FileBasedUserHome (changes, review)

VerifierTask (changes)

  VerifierTask.execute() (crap4j)

FileBasedContentTracker (review, bugs)

  FileBasedContentTracker.fsck() (crap4j)

  FileBasedContentTracker.gc() (crap4j)

index.jsp (changes)

HarversterTask (bugs)

Now we know where to start, and now let's talk about how to start.

Gaining 30% coverage in 2 minutes:

```
public class CheaterTest {
    @Test
    public void shouldIncreaseTheCoverage() {
        HarvesterTask harvester = new HarvesterTask();
        Project project = new Project();
        project.setBaseDir(new File("."));
        harvester.setProject(project);
        harvester.setRepository("../repository");
        harvester.setHistory("history");
        harvester.setTemplate("templates");
        harvester.execute();
    }
}
```

Covered code != Tested code

So, you start with an assertion:

```
public class FileHelperTest {
    @Test
    public void shouldReturnTheContentOfAFile() throws IOException {
        assertEquals("", FileHelper.getFileContent(null));
    }
}
```

➡ The ‘assertEquals’ makes sure that your test actually does something

➡ The ‘null’ parameter - along with the NullPointerException - will show you where to continue

First test case is done:

```
public class FileHelperTest {
    @Test
    public void shouldReturnTheContentOfAFile() throws IOException {
        File input = File.createTempFile("foo", "bar");
        assertEquals("", FileHelper.getFileContent(input));
    }
}
```

➡ Now the test is green, let's continue with a more meaningful test case

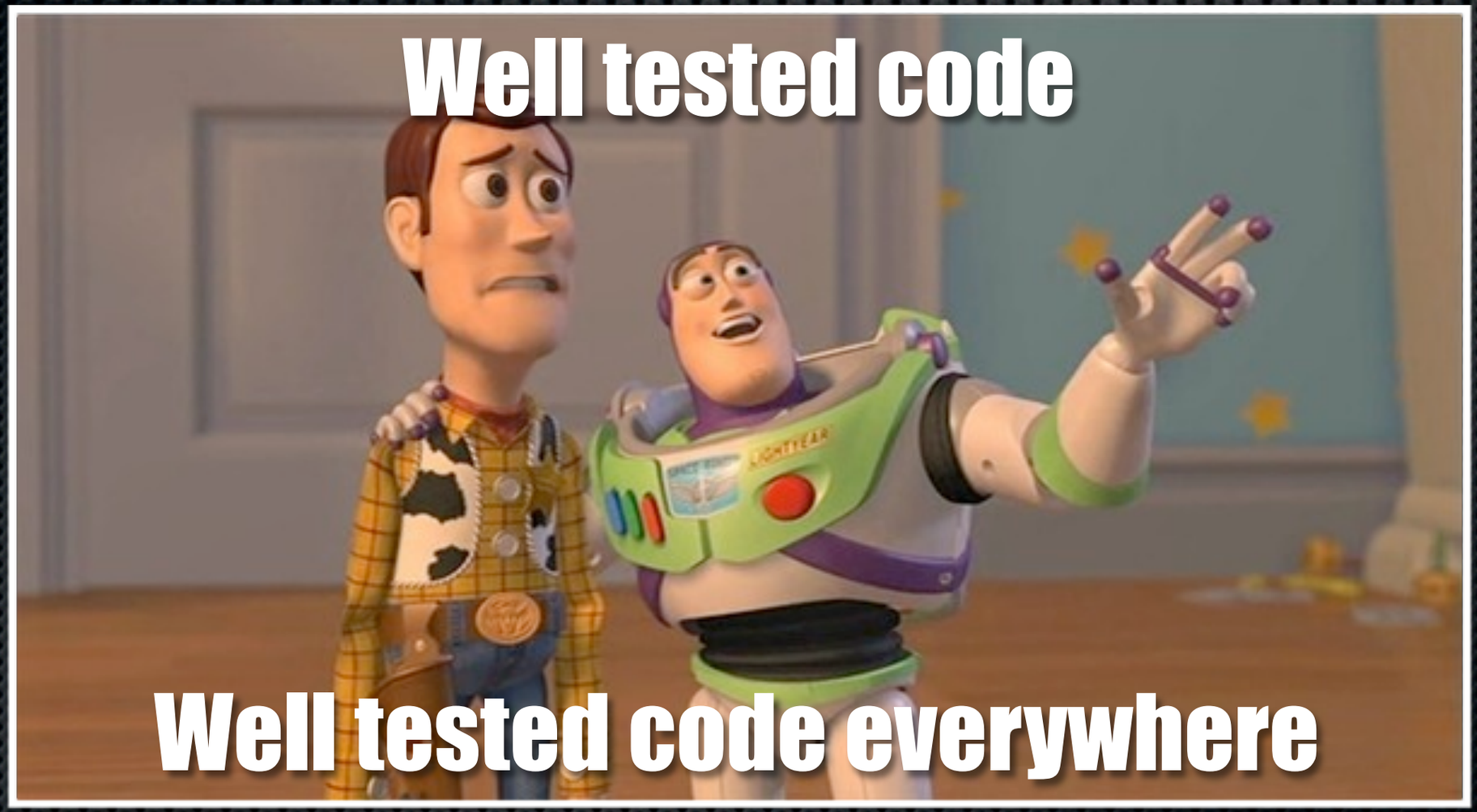
Now we have two test cases:

```
public class FileHelperTest {
    @Test
    public void shouldReturnTheContentOfAFile() throws IOException {
        File input = File.createTempFile("foo", "bar");
        assertEquals("", FileHelper.getFileContent(input));
    }

    @Test
    public void shouldReturnTheContentOfAFile() throws IOException {
        File input = File.createTempFile("foo", "bar");
        new FileOutputStream(input).write("something".getBytes());
        assertEquals("something", FileHelper.getFileContent(input));
    }
}
```

➡ Test method names remains the same until the body is filled properly

And we are done (assertion + coverage):

```
public class FileHelperTest {
    private File input;

    @Before
    public void setUp() throws IOException {
        input = File.createTempFile("foo", "bar");
    }

    @Test
    public void shouldReturnAnEmptyStringForAnEmptyFile() throws IOException {
        assertEquals("", FileHelper.getFileContent(input));
    }

    @Test
    public void shouldReturnTheContentOfAFile() throws IOException {
        setInputFileContent("something");
        assertEquals("something", FileHelper.getFileContent(input));
    }

    private void setInputFileContent(String content) throws IOException {
        new FileOutputStream(input).write("something".getBytes());
    }
}
```


Well tested code
Well tested code everywhere

What about web applications?

(I'll use a ruby on rails example, but the principles apply to other frameworks as well)

Points

#2 Find out which parts of the code change often (a.k.a VCS statistics)

and

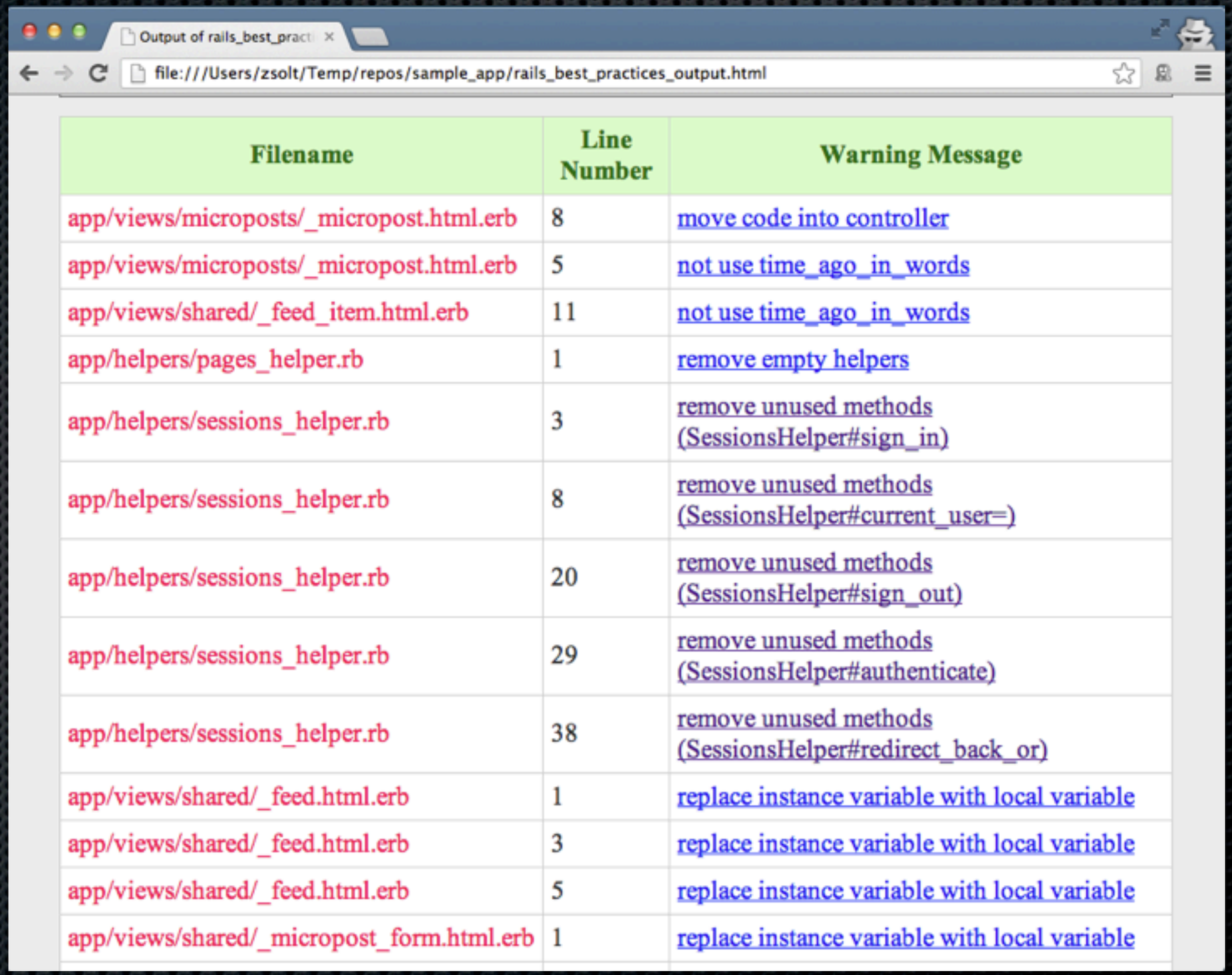
#3 Determine which part of the code changes data (a.k.a code review)

are just the same.

A large variety of tools are available for:

#4 Determine where the code is most likely going to fail (a.k.a static code checkers)

```
% gem install rails_best_practices
% rails_best_practices -html .
```



Filename	Line Number	Warning Message
app/views/microposts/_micropost.html.erb	8	move code into controller
app/views/microposts/_micropost.html.erb	5	not use time_ago_in_words
app/views/shared/_feed_item.html.erb	11	not use time_ago_in_words
app/helpers/pages_helper.rb	1	remove empty helpers
app/helpers/sessions_helper.rb	3	remove unused methods (SessionsHelper#sign_in)
app/helpers/sessions_helper.rb	8	remove unused methods (SessionsHelper#current_user=)
app/helpers/sessions_helper.rb	20	remove unused methods (SessionsHelper#sign_out)
app/helpers/sessions_helper.rb	29	remove unused methods (SessionsHelper#authenticate)
app/helpers/sessions_helper.rb	38	remove unused methods (SessionsHelper#redirect_back_or)
app/views/shared/_feed.html.erb	1	replace instance variable with local variable
app/views/shared/_feed.html.erb	3	replace instance variable with local variable
app/views/shared/_feed.html.erb	5	replace instance variable with local variable
app/views/shared/_micropost_form.html.erb	1	replace instance variable with local variable

https://codeclimate.com

Classes - ZsoltFabok/site_

https://codeclimate.com/github/ZsoltFabok/site_checker/constants

Raise the visiblity of code quality in your company with Code Climate. Learn More

CODE CLIMATE

Learn More Login

ZsoltFabok/site_ch...

Feed | Classes | Smells | Trends

Search by class name

code climate 3.8

Tweet

dd1cc547 g

Rating	Class Name	Complexity	Duplication	Churn	Methods	C/M	Smells
A	SiteChecker	39	0	10	8	4.8	0
B	SiteChecker::Cli	125	0	2	6	20.9	2
A	SiteChecker::DSL	13	0	1	0	*N/A*	0
A	SiteChecker::IO	0	0	1	0	*N/A*	0
A	SiteChecker::IO::ContentFromFileSystem	31	0	1	4	7.7	0
A	SiteChecker::IO::ContentFromWeb	22	0	1	3	7.4	0
A	SiteChecker::Link	42	0	3	13	3.3	0
A	SiteChecker::LinkCollector	118	0	4	19	6.2	0

Feed – Code Climate

https://codeclimate.com/github/rails/rails

B → A ActionView::Helpers::Tags::Base has improved. a day ago

C → A ActionView::Template::Handlers::ERB has improved. a day ago

A → B ActionView::Template::Handlers::Erubis has gotten worse. a day ago

Three classes/modules were added. 4 days ago

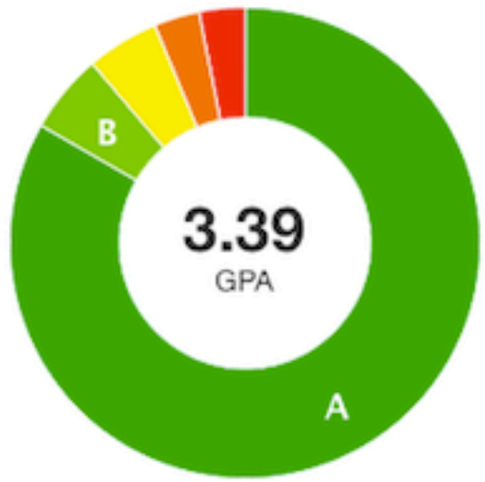
- A ActiveRecord::Tasks::FirebirdDatabaseTasks
- A ActiveRecord::Tasks::OracleDatabaseTasks
- A ActiveRecord::Tasks::SqlserverDatabaseTasks

F → D ActiveRecord::FinderMethods has improved. 5 days ago

A → B ActionDispatch::Cookies::CookieJar has gotten worse. 5 days ago

A → C Rails::VERSION has gotten worse. 5 days ago

Classes by Rating



Hotspots

- F HTML::Selector
- F Rails::Engine
- F Time
- F ActionDispatch::Assertions::SelectorAssertion
- F ActiveSupport::Multibyte::Unicode

Everything is nice and straightforward
until now, but the last remaining point
is tricky:

#1 Determine which parts of the code
are really used (a.k.a. coverage)

We can have coverage data in Ruby on Rails, too:

```
~/Temp/repos/sample_app % gem install simplecov
~/Temp/repos/sample_app % cat script/rails
#!/usr/bin/env ruby

require 'simplecov'
SimpleCov.start do
  add_group "Models", "app/models"
  add_group "Controllers", "app/controllers"
end

APP_PATH = File.expand_path('../../config/application',  __FILE__)

# rest of the script/rails script
```

Code coverage for Sample

file:///Users/zsolt/Temp/repos/sample_app/coverage/index.html#_AllFiles

All Files (80.33%) | Models (80.7%) | Controllers (74.19%) | Ungrouped (82.11%)

Generated 13 days ago

All Files (80.33% covered at 0.96 hits/line)

21 files in total. 183 relevant lines. 147 lines covered and 36 lines missed

Search:

File	% covered	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
app/helpers/sessions_helper.rb	62.5 %	61	32	20	12	1.2
config/boot.rb	66.67 %	13	9	6	3	0.7
app/controllers/microposts_controller.rb	68.75 %	26	16	11	5	0.7
app/models/user.rb	72.5 %	93	40	29	11	0.7
app/controllers/pages_controller.rb	75.0 %	22	12	9	3	0.8
app/helpers/application_helper.rb	87.5 %	16	8	7	1	0.9
app/helpers/microposts_helper.rb	88.89 %	15	9	8	1	3.3
app/controllers/application_controller.rb	100.0 %	4	3	3	0	1.0
app/helpers/pages_helper.rb	100.0 %	2	1	1	0	1.0
app/helpers/users_helper.rb	100.0 %	7	3	3	0	1.7
app/models/micropost.rb	100.0 %	33	11	11	0	1.1
app/models/relationship.rb	100.0 %	21	6	6	0	1.0
config/application.rb	100.0 %	42	7	7	0	1.0
config/environment.rb	100.0 %	5	2	2	0	1.0
config/environments/development.rb	100.0 %	22	8	8	0	1.0
config/initializers/backtrace_silencers.rb	100.0 %	7	0	0	0	0
config/initializers/inflections.rb	100.0 %	10	0	0	0	0
config/initializers/mime_types.rb	100.0 %	5	0	0	0	0
config/initializers/secret_token.rb	100.0 %	7	1	1	0	1.0
config/initializers/session_store.rb	100.0 %	8	1	1	0	1.0
config/routes.rb	100.0 %	78	14	14	0	1.0

There is only one problem: the application must be stopped in order to get the report, which is not really efficient and user friendly.

Fortunately, we can use a metric called ‘funnel’:

Goal Funnel - Google Anal

https://www.google.com/analytics/web/?hl=en&pli=1#report/conversions-goal-funnel/a16...

www.zsoltfabok.com - http://www.zsoltfabok.com
zsoltfabok.com [DEFAULT]

Reporting | Customization | Adm

176

176

160

(entrance)	166
[redacted]	2
[redacted]	1
[redacted]	1
[redacted]	1

1. This is a huge lost in visitors, would be good to know why.

32

16

22

(entrance)	12

(exit)	14
[redacted]	1
[redacted]	1
[redacted]	1
[redacted]	1

2. Users visit this page more often than the other page.

funnel
78

68

30.00% funnel conversion rate

(entrance)	56
[redacted]	2

Slides: http://zsoltfabok.com/speaking/

Code: https://github.com/ZsoltFabok/arithmetic.expression.evaluator/tree/xp2013

Thank you very much for your attention!

http://zsoltfabok.com/

@ZsoltFabok