SUNGARD GLOBAL SERVICES

Development Tools for Large Teams

September 8, 2012

Todd Ricker

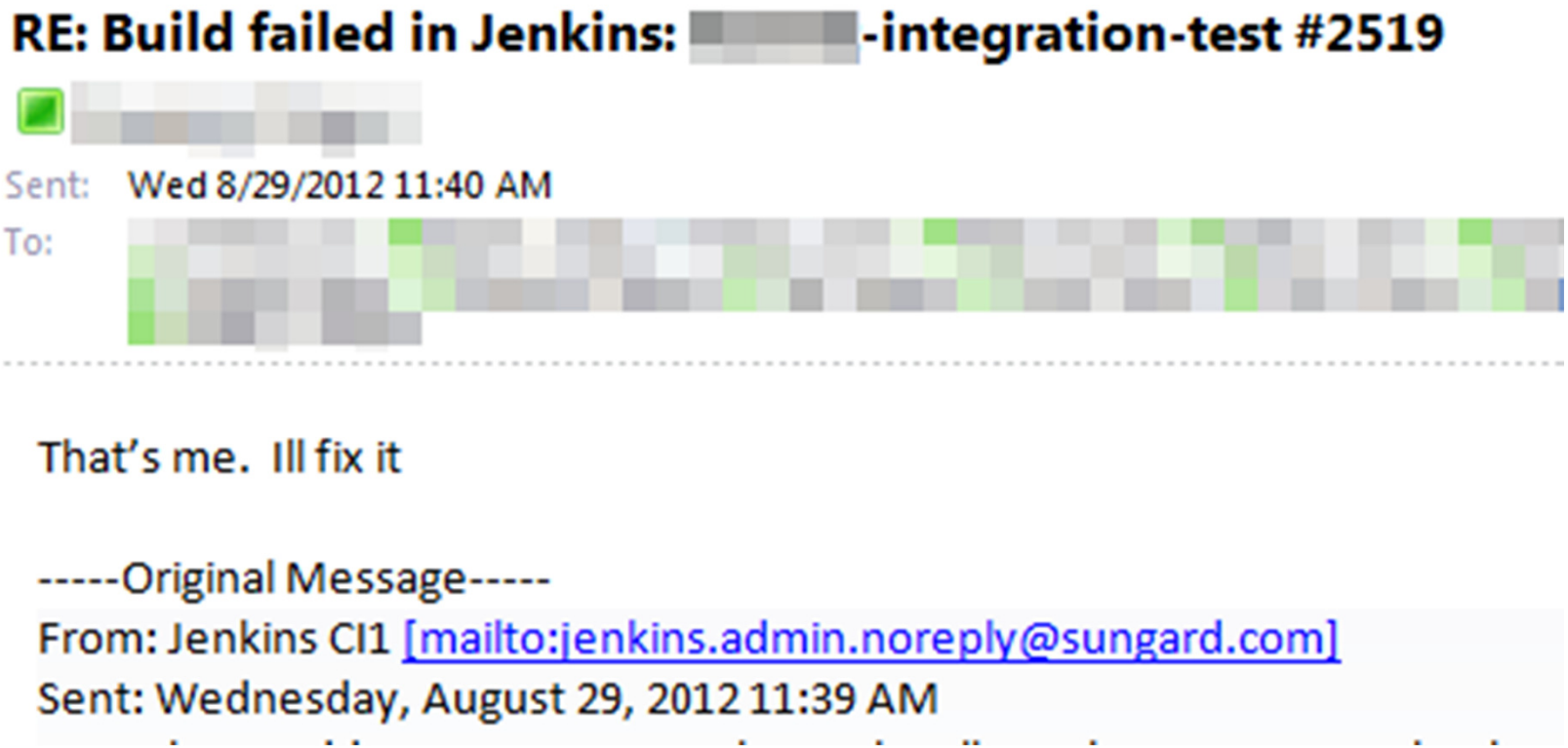
A few questions

- Other than implementing application functionality, where do your developers spend their time?
- What is the quality of your team's code? How brittle is it?
- When things break, how long does it take the team to know something is broken?
- When things break, do people self-mobilize to fix the problem?
- Can your QA staff do their own builds?
- Do you have a one click build?

Goals: Agile / Continuous integration / Devops

- Eliminate manual processes
- Stringent and continual testing
- Frequent, low-ceremony releases
- Cultural shift to shared accountability
- Transparency

Shared accountability: An example



RE: Build failed in Jenkins: [redacted]-integration-test #2519

[redacted]

Sent: Wed 8/29/2012 11:40 AM

To: [redacted]

That's me. Ill fix it

-----Original Message-----

From: Jenkins CI1 [mailto:jenkins.admin.noreply@sungard.com]

Sent: Wednesday, August 29, 2012 11:39 AM

Solutions

- Continuous integration
  - Continuously compile, test, and deploy
- Software quality tools
- Unit test coverage tools
- Javadoc coverage tools
- Peer code reviews
- Automated functional tests
- Database change provisioning

Devops

- The tools and processes tread the line between many roles
  - Development
  - Systems administrations
  - QA / Testing staff
  - Build engineers

Tools: Continuous integration

- What is it?
  - Short answer: Build (and test) automation
  - The practice of frequently integrating one's new or changed code with the existing code repository – should occur frequently enough that no intervening window remains between commit and build, and such that no errors can arise without developers noticing them and correcting them immediately

Source: Wikipedia.org

http://en.wikipedia.org/wiki/Continuous_integration

Tools: Continuous integration

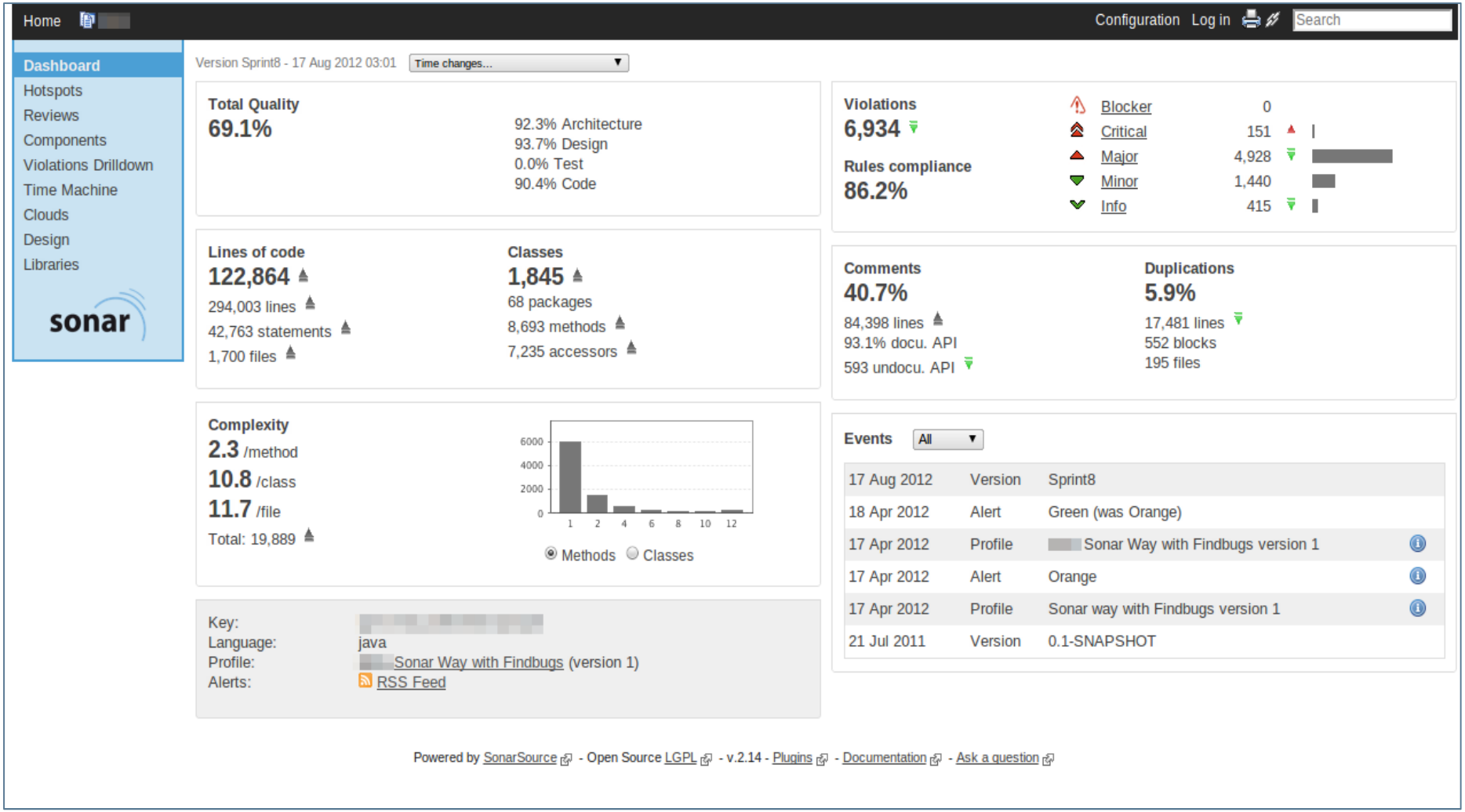
- Jenkins (Open Source)
- Hudson (Open Source)
- CruiseControl (Open Source)
- JetBrains TeamCity
- Atlassian Bamboo
- And many others...

Live Jenkins demo
coming up...

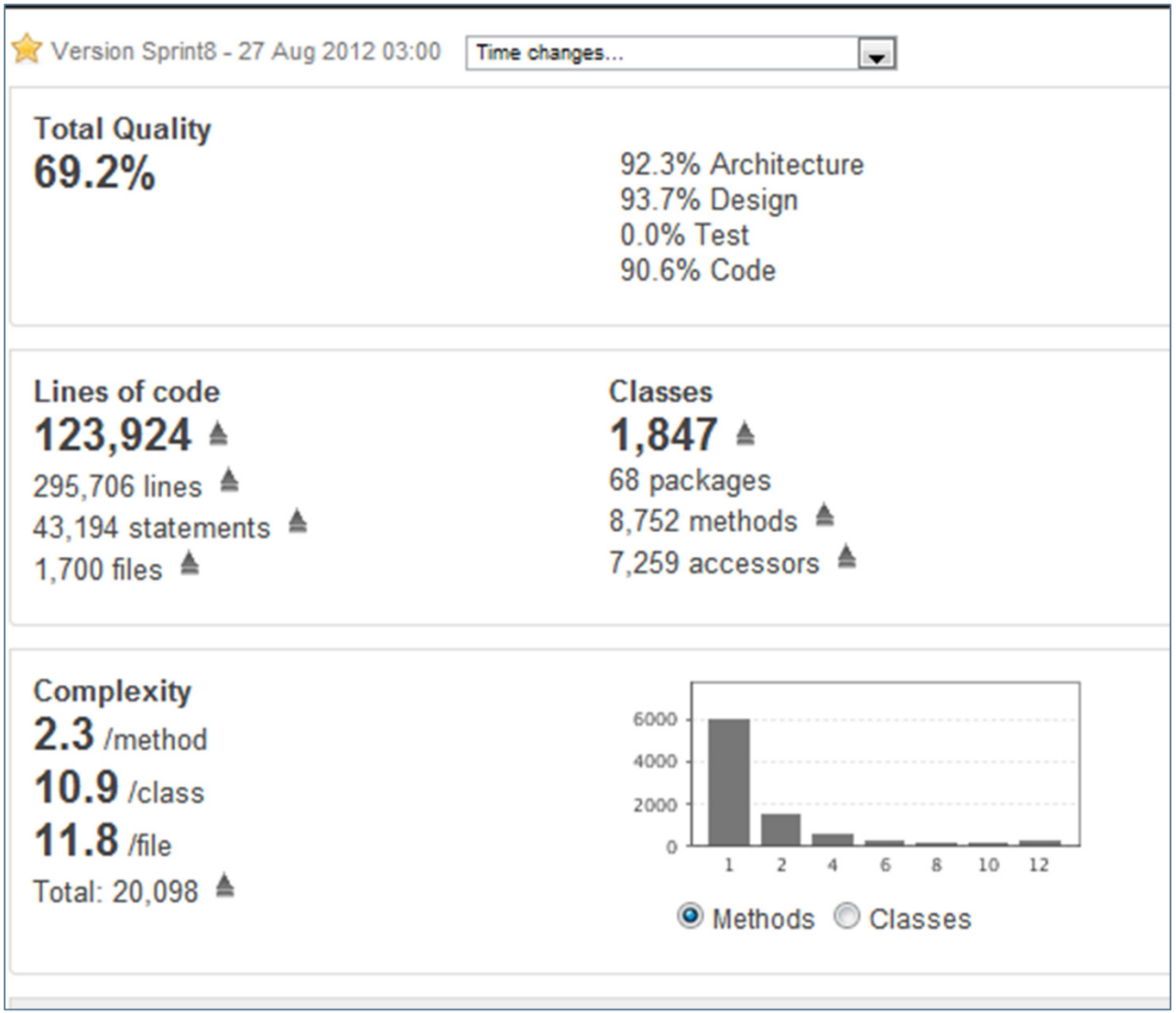
Tools: Software quality – Sonar

- "Sonar is an open platform to manage code quality"
  source: http://www.sonarsource.org/
- The 7 axes of code quality
  - Architecture & design
  - Duplications
  - Unit tests
  - Complexity
  - Potential bugs - http://findbugs.sourceforge.net/
  - Code rules - http://checkstyle.sourceforge.net/
  - Comments

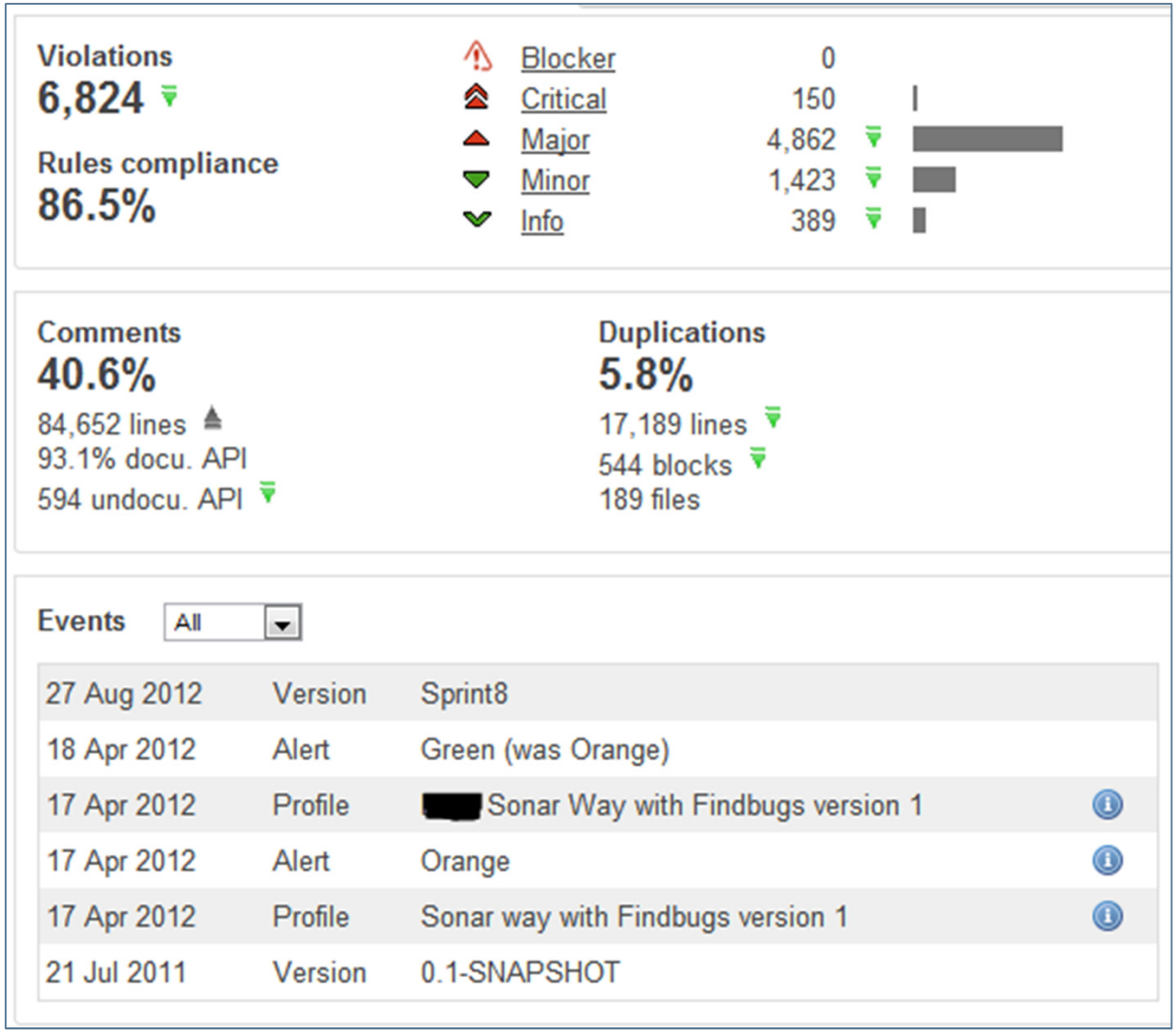
Tools: Sonar – Project overview page



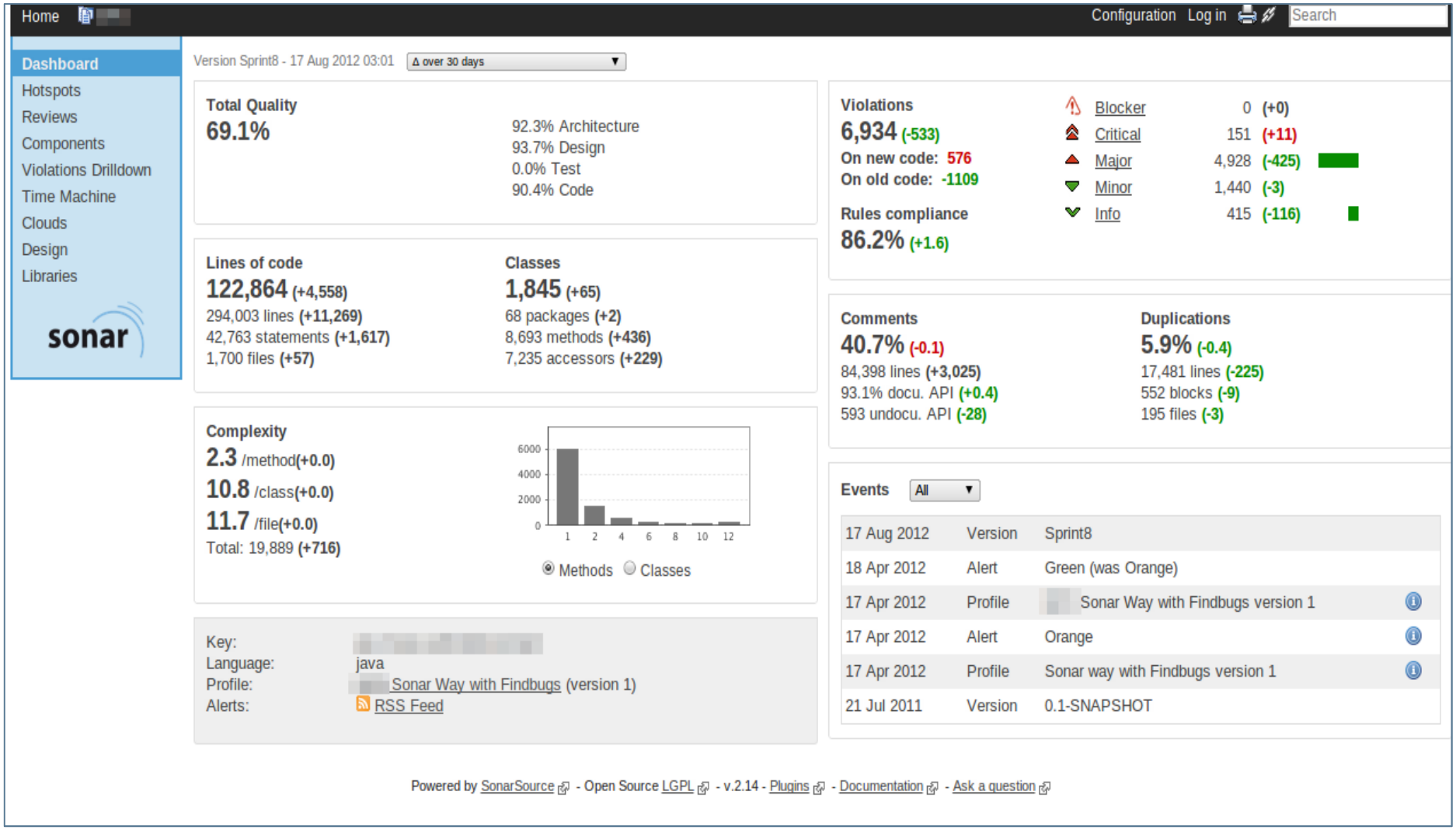
Tools: Sonar – Project overview page



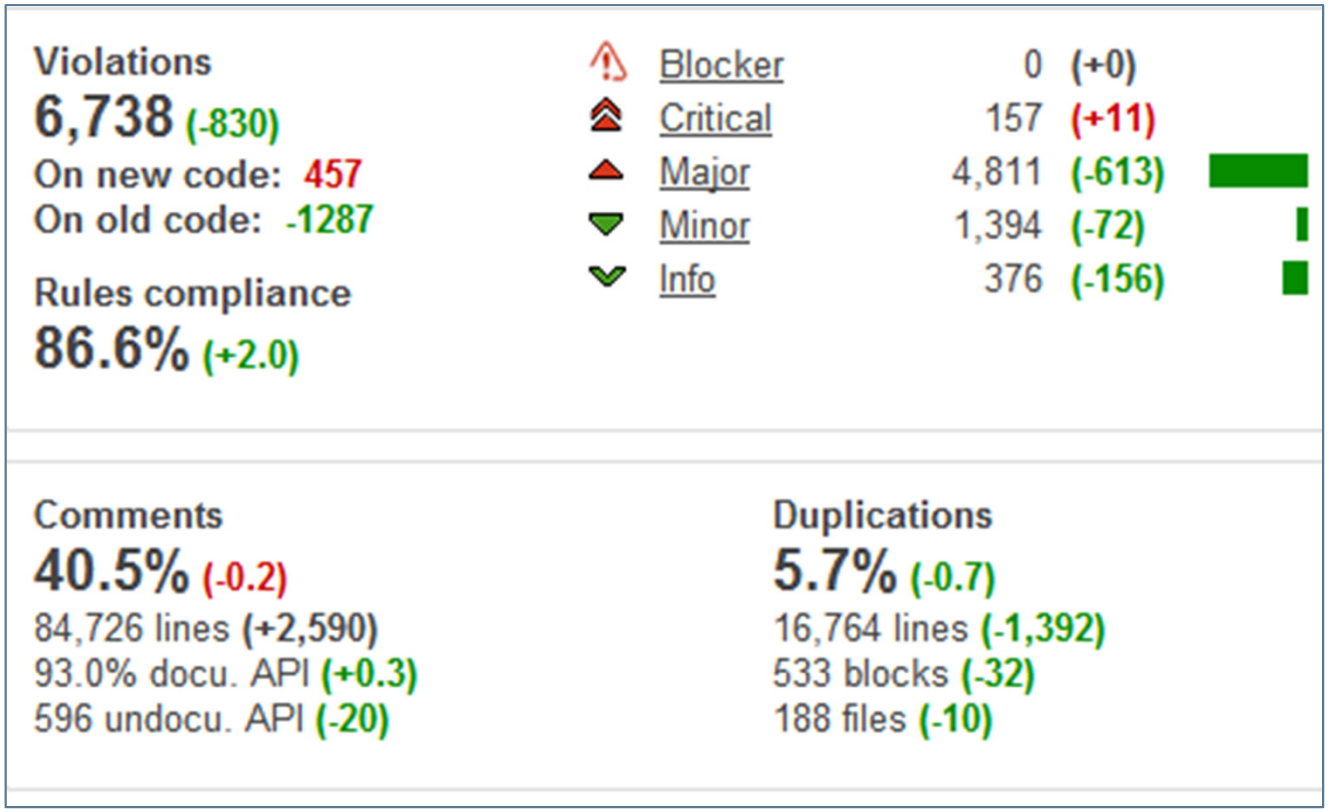
Tools: Sonar – Project overview page



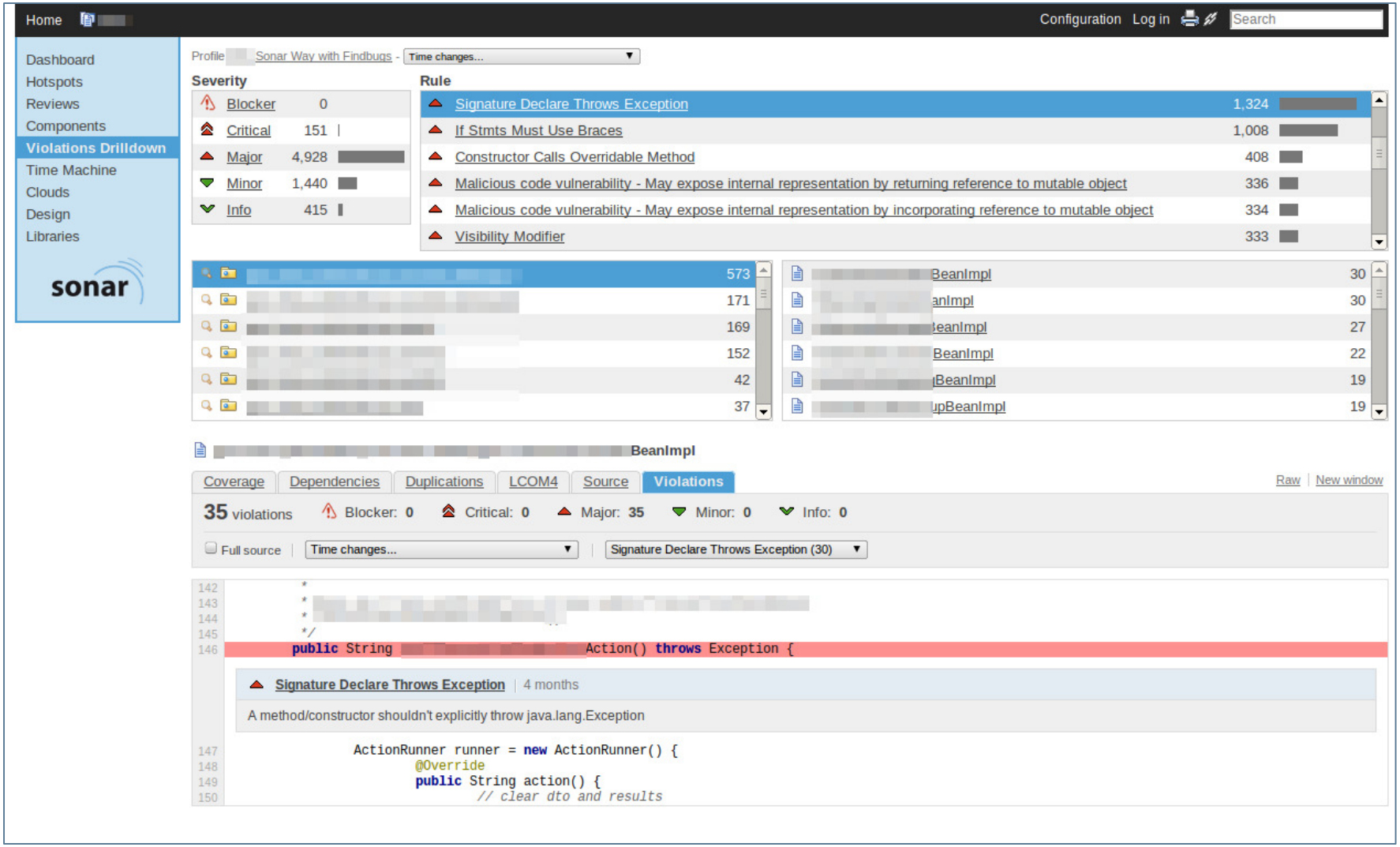
Tools: Sonar – 30 Day Delta



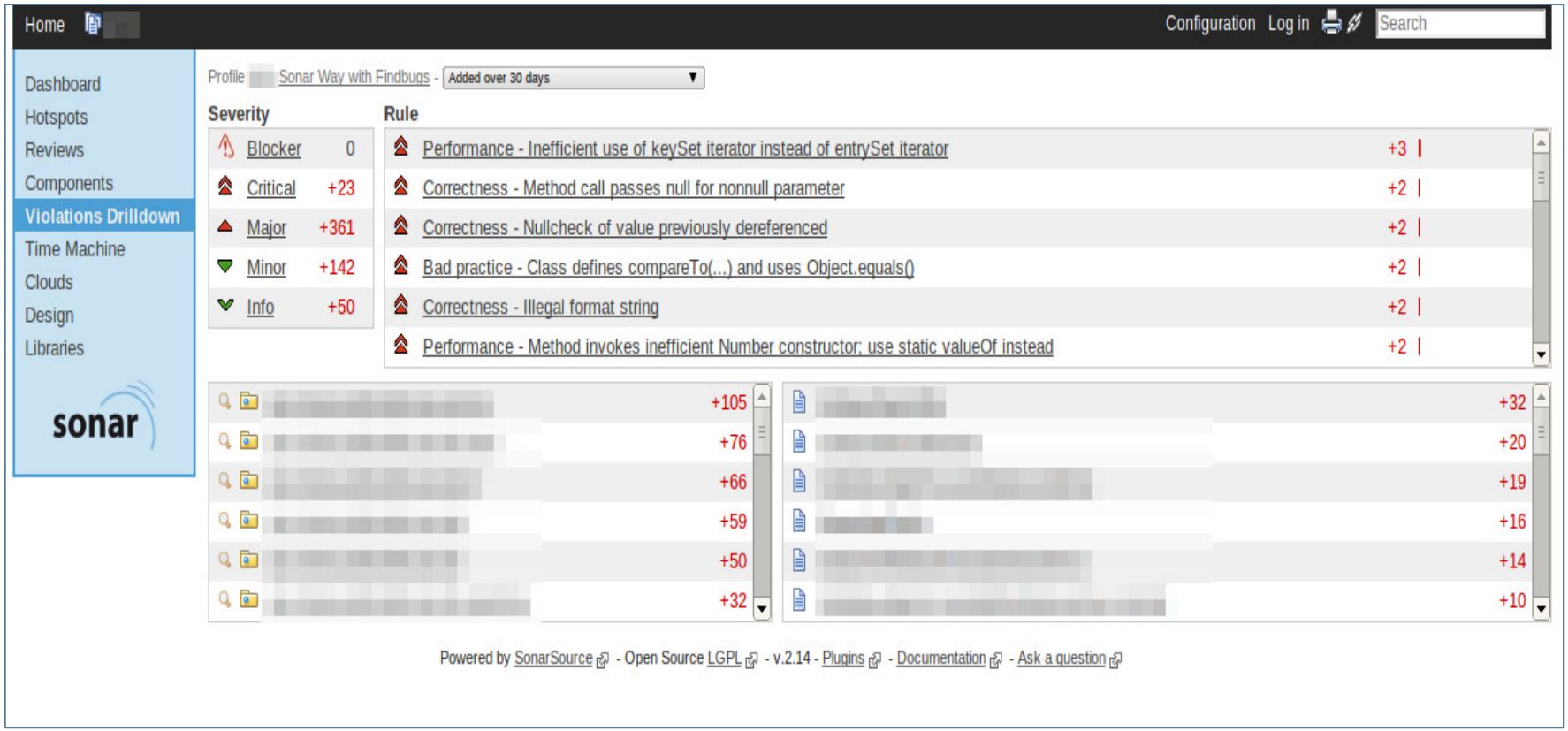
Tools: Sonar – 30 Day Delta



Tools: Sonar – Violations

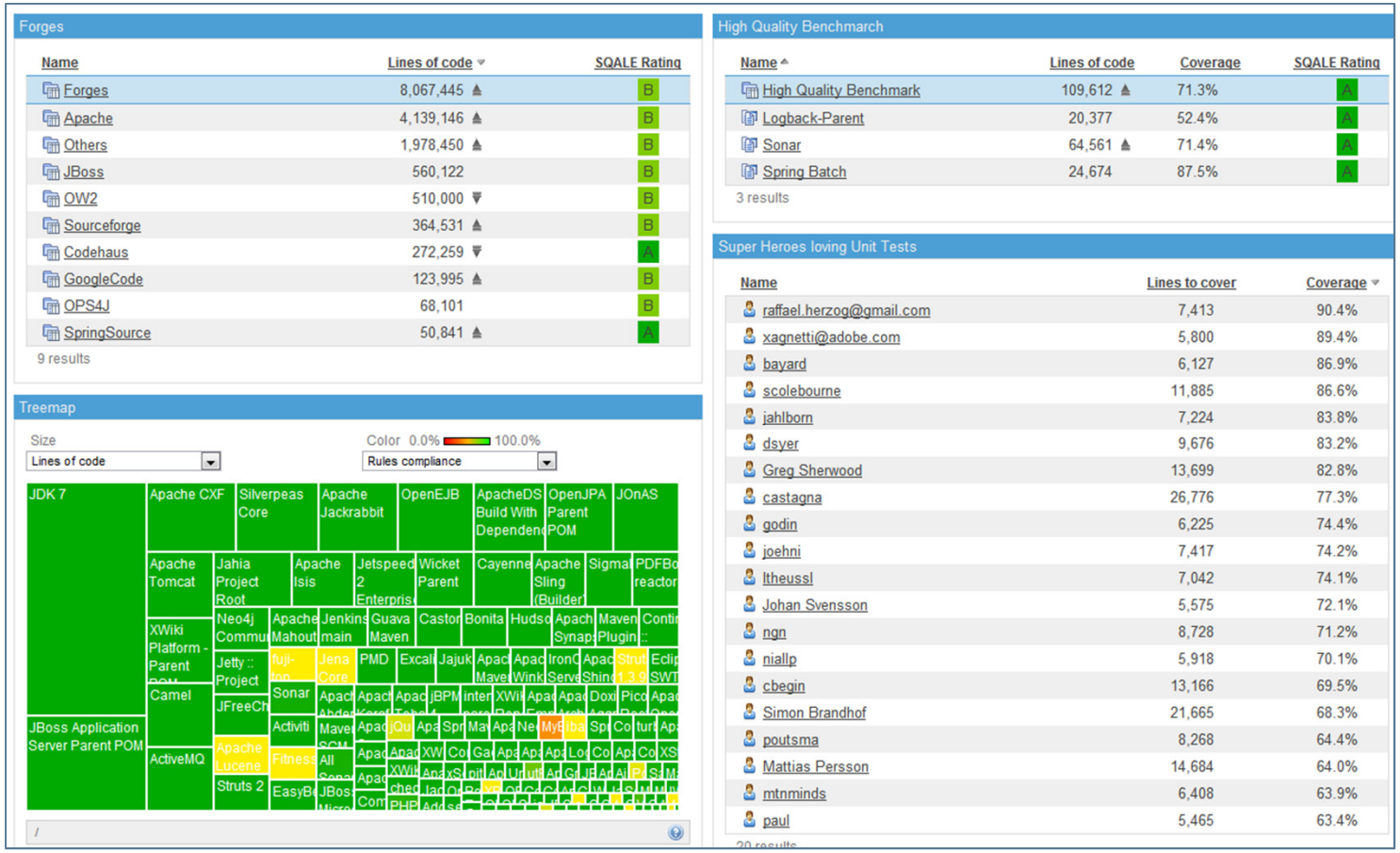


Tools: Sonar – Violations – 30 Day Delta



Tools: Sonar

- Live Sonar instance tracking many OSS projects:
  - http://nemo.sonarsource.org/



Tools: Unit test coverage

- ***"I don't think anybody tests enough of anything"*** – A Conversation with Java's Creator, James Gosling, by Bill Venners, 2002
  http://www.artima.com/intv/goslingD4.html
- In a large codebase it is virtually impossible to "manually" gauge Unit test coverage
- Options
  - Cobertura – Java, (Open Source)
  - Atlassian Clover – Java, Groovy
  - Ncover – .NET

Tools: Unit test coverage – Cobertura

- How does it work?
  - 3 Steps
    1. Instrumentation - added to your Java byte-code
    2. Execute unit test
    3. Generate html report

Tools: Unit Test Coverage – Cobertura

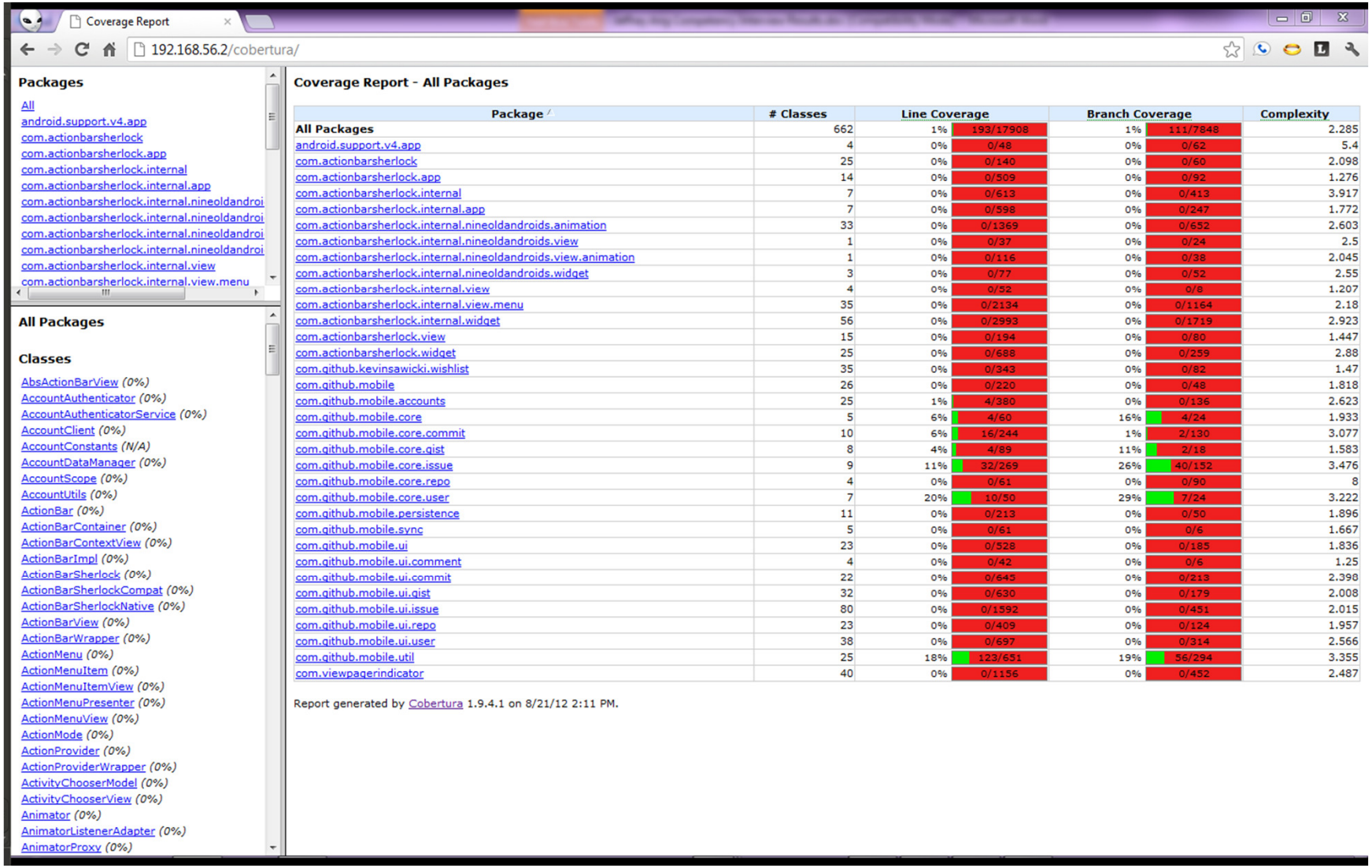
- Easy configuration with Maven

```
<reporting>
  <plugins>
    <plugin>
      <groupId>org.codehaus.mojo</groupId>
      <artifactId>cobertura-maven-plugin</artifactId>
      <version>2.5.1</version>
      <configuration>
        <instrumentation/>
      </configuration>
    </plugin>
  </plugins>
</reporting>
```

- To run with maven:

  mvn cobertura:cobertura

Tools: Unit Test Coverage - Cobertura



Tools: Unit Test Coverage - Cobertura

```
92                  /**
93                   * Add HTML colorization to a block of Java code.
94                   *
95                   * @param text The block of Java code.
96                   * @return The same block of Java code with added span tags.
97                   *         Newlines are preserved.
98                   */
99                  public String process(final String text)
100                 {
101 500                     if (text == null)
102   0                         throw new IllegalArgumentException("\"text\" can not be null.");
103
104 500                     StringBuffer ret = new StringBuffer();
105
106                     // This look is really complicated because it preserves all
107                     // combinations of \r, \n, \r\n, and \n\r
108                     int begin, end, nextCR;
109 500                 begin = 0;
110 500                 end = text.indexOf('\n', begin);
111 500                 nextCR = text.indexOf('\r', begin);
112 500                 if ((nextCR != -1) && ((end == -1) || (nextCR < end)))
113   0                     end = nextCR;
114 500                 while (end != -1)
115                     {
116   0                     ret.append(processLine(text.substring(begin, end)) + "<br/>");
117
118   0                     if ((end + 1 < text.length())
119                                 && ((text.charAt(end + 1) == '\n') || (text
120                                         .charAt(end + 1) == '\r')))
121                         {
122   0                         ret.append(text.substring(end, end + 1));
123   0                         begin = end + 2;
124   0                     }
125                         else
126                         {
127   0                         ret.append(text.charAt(end));
128   0                         begin = end + 1;
129                         }
130
131   0                     end = text.indexOf('\n', begin);
132   0                     nextCR = text.indexOf('\r', begin);
133   0                     if ((nextCR != -1) && ((end == -1) || (nextCR < end)))
134   0                         end = nextCR;
135                     }
136 500                 ret.append(processLine(text.substring(begin)));
137
138 500                 return ret.toString();
139                 }
140
```

JavaDoc coverage

- Doc check
  - Extension of Javadoc tool from Sun / Oracle
  - Generates an html report documenting Javadoc coverage levels and specific areas with omissions
- *A good idea for consultants delivering new APIs to clients*

JavaDoc coverage

API Specification Errors Executive Summary for DocCheck Sample Code

Generated by DocCheck doclet, version 1.2b2,
Fri, 11 Jul 2003.

For a summary by package, see Package Summary
For a statistical breakdown, see Package Statistics

Missing Comments

Package	Number	Percent
All Packages	122	49%
testPkg1.*	40	51%
testPkg2.*	82	53%

Minor Errors Only

Package	Number	Percent
All Packages	67	27%
okpkg.*	1	33%

JavaDoc coverage

API Specification Errors
Errors in Package: testPkg1

Package

Category 1: Package Error

- No documentation for package. Need a **package.html** file.

Class/Interface

	Missing Comments		Minor Errors		
Name	**Class/Iface**	**Member**	**Tag**	**Text/Link**	**Warning**
Test1	1	19	12	8	--
Test2	--	19	12	8	--

Explanation of Terms

Member

An inner class, inner interface, field, constructor, or method.

Note:

This table shows total numbers for the minor errors, rather than the number of items that have such an error. (If there was one method missing 4 tags, this table would show "4" in the Tags column.)

Automated functional testing

- Selenium 2.0 – WebDriver API
- Improvement over 1.0; no longer uses JavaScript trickery to drive the browsers; now uses native browser calls
- Drivers for Firefox, Chrome, IE, Safari, *and even Android*
- Tests can be written in Java, Ruby, C#, Python, Perl, PHP

Automated functional testing

- *I choose the Ruby WebDriver API because:*
  - It can run “headless” in a CI environment
  - Headless - Ruby gem for Xvfb, a lightweight virtual framebuffer
- And for the fact that I can write my tests in Ruby

Automated functional testing - Simplest

```ruby
require "selenium-webdriver"

driver = Selenium::WebDriver.for :firefox
driver.navigate.to "http://google.com"

element = driver.find_element(:name, 'q')
element.send_keys "Hello WebDriver!"
element.submit

puts driver.title

driver.quit
```

Source: http://code.google.com/p/selenium/wiki/RubyBindings

Automated functional testing – Advanced

```ruby
#!/usr/bin/env ruby

require 'selenium-webdriver'
require 'headless'
require 'test/unit'

module TestHelper

  Runheadless = true

  def whoami
        "#{self.class.name}"
  end

  def setup
        if Runheadless
                @headless = Headless.new
                @headless.start
        end

        @driver = Selenium::WebDriver.for :firefox
        @driver.manage.timeouts.implicit_wait = 30
        @verification_errors = []
        @errors = true
  end
```

Automated functional testing - Advanced

```
def teardown
  if @errors
      t = Time.now

      @driver.save_screenshot(t.strftime("hrnt-#{whoami}-fail-%m%d%Y-%H%M.png"))
      @driver.quit
      assert false;
  end
  @driver.quit
  if Runheadless
      @headless.destroy
  end
  assert_equal [], @verification_errors
end

def verify(&blk)
  yield
  rescue Test::Unit::AssertionFailedError => ex
  @verification_errors << ex
end
```

Code Reviews

- The benefits
  - Improved Quality
  - Professional Development
- The challenges
  - Logistics, scheduling, out sick, conference rooms, etc.
  - Keeping the tone and scope of reviews constructive
  - Making it “automatic”

Code reviews

- Make it part of your “culture”
  - Do them on a set schedule and make sure they happen
  - Assign the reviewee as the “host”, responsible for the meeting logistics
  - Set the expectation that reviewers come prepared
- Use a tool!
  - Atlassian Crucible – proprietary - any version control system
  - Gerrit – Open Source – git based
    - Many others…

Simple code review framework

- 1 reviewee, 2 reviewers
- Reviewee sets up review, pulling in any code in question
- Reviewers have a few days to review code at their leisure. Comments are added via the tool.
- Reviewee and reviewers get together face to face and go through the code and address all comments
- Document and prioritize any to-do items that come out of the review

Development tools: Database provisioning

- Liquibase
  - "…an open source (Apache 2.0 Licensed), database-independent library for tracking, managing and applying database changes."
    source: http://www.liquibase.org/
  - Store database structure and updates in your version control system
  - Replay DDL changes and rebuild at will. Apply only most recent changesets.
  - Push to any flavor of database using Java JDBC drivers
  - We develop locally on MySQL and push to integration and production DB2 database servers

Development tools – Database provisioning

- Liquibase

```xml
<?xml version="1.0" encoding="UTF-8"?>

<databaseChangeLog
  xmlns="http://www.liquibase.org/xml/ns/dbchangelog"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog
         http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-2.0.xsd">

    <changeSet id="1" author="bob">
        <createTable tableName="department">
            <column name="id" type="int">
                <constraints primaryKey="true" nullable="false"/>
            </column>
            <column name="name" type="varchar(50)">
                <constraints nullable="false"/>
            </column>
            <column name="active" type="boolean" defaultValueBoolean="true"/>
        </createTable>
    </changeSet>

</databaseChangeLog>
```

source: http://www.liquibase.org/quickstart

Development tools – Database provisioning

- Liquibase

```
liquibase --driver=com.mysql.jdbc.Driver \
     --classpath=/path/to/classes \
     --changeLogFile=com/example/db.changelog.xml \
     --url="jdbc:mysql://localhost/example" \
     --username=user \
     --password=asdf \
     migrate
```

source: http://www.liquibase.org/quickstart

- Or better yet, configure maven: mvn liquibase:update
- We use one changelog per sprint

Development tools – Database provisioning

» Liquibase – loading CSV spell out reference data

```
<loadData tableName="users" file="com/sample/users.csv">
    <column name="id" type="NUMERIC"/>
    <column name="firstname" type="STRING"/>
    <column name="lastname" type="STRING"/>
    <column name="username" type="STRING"/>
</loadData>
```

source: http://www.liquibase.org/manual/load_data

What is Jenkins?

- A continuous integration system!
  - Everything discussed thus far should be automated with Jenkins jobs

Jenkins continuous integration

- Install
  - Download jenkins.war
  - java –jar jenkins.war
  - (Or use Apache Tomcat)
- Demo: http://192.168.56.2:8080/

SUNGARD GLOBAL SERVICES

Contact

Todd Ricker
Senior Manager
SunGard Global Services
340 Madison Ave., 7th Floor
New York, NY 10173

+1 917 727 8633 phone
Todd.ricker@sungard.com