Sonar

10 Metrics for Better Builds
<table>
<thead>
<tr>
<th>Date</th>
<th>Title</th>
<th>Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>28 Nov 10</td>
<td><strong>Sonar</strong> » Sonar has become a Multi-Languages Platform</td>
<td>jvm, language, sonar</td>
</tr>
<tr>
<td>25 Nov 10</td>
<td><strong>Metric definitions - Sonar - Codehaus</strong></td>
<td>metrics, statistics, sonar</td>
</tr>
<tr>
<td></td>
<td><strong>Install Sonar - Sonar - Codehaus</strong></td>
<td>sonar</td>
</tr>
<tr>
<td></td>
<td><strong>Sonar in a nutshell - Sonar - Codehaus</strong></td>
<td>sonar</td>
</tr>
<tr>
<td>19 Nov 10</td>
<td><strong>Sonatype Blog » Maven 3 and Sonar</strong></td>
<td>continuousintegration, maven, maven3, sonar</td>
</tr>
<tr>
<td></td>
<td>[#GRADLE-888] Gradle Integration with Sonar - jira.codehaus.org</td>
<td>gradle, continuousintegration, sonar</td>
</tr>
<tr>
<td></td>
<td><strong>Sonar » Sonar in the news</strong></td>
<td>continuousintegration, newsevent, sonar</td>
</tr>
<tr>
<td></td>
<td>[sonar-dev] Sonar 2.5 will start answering the question &quot;What has changed ?&quot; - Freddy Mallet - org.codehaus.sonar.dev - MarkMail</td>
<td></td>
</tr>
</tbody>
</table>

http://delicious.com/matthew.mccullough/sonar
What is this?

Measurements are for managers
measurements are for managers
true...
and also for developers
sonar
Quality is under control
what measurements?
★ Checkstyle
★ PMD
★ Findbugs
★ Cobertura
★ Emma
★ Clirr
★ JaCoCo
★ Useless Code
★ SQALE
★ 20+ others...
very little plaintext...
Test set: org.junit.samples.SimpleTest

Tests run: 1, Failures: 1, Errors: 0, Skipped: 0, Time elapsed: 0.002 sec <<< FAILURE!
org.junit.samples:SimpleTest.testEquals() Time elapsed: 0 sec <<< FAILURE:
  java.lang.AssertionError: Size expected:<12> but was:<13>
  at org.junit.Assert.failTestAssertion(Assert.java:91)
  at org.junit.Assert.failNotEquals(Assert.java:645)
  at org.junit.Assert.assertEquals(Assert.java:126)
  at org.junit.Assert.assertEquals(Assert.java:470)
  at org.junit.samples:SimpleTest.testEquals(SimpleTest.java:35)
instead, graphs
<table>
<thead>
<tr>
<th>Alert</th>
<th>Name</th>
<th>Version</th>
<th>Lines of code</th>
<th>Rules compliance</th>
<th>Build date</th>
<th>Links</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sample Multimodule Java Parent</td>
<td>1.0-SNAPSHOT</td>
<td>48</td>
<td>25.0%</td>
<td>2010-02-05</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sample Project - Algorithms</td>
<td>1.0-SNAPSHOT</td>
<td>8</td>
<td>25.0%</td>
<td>2010-10-26</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sample Project - Clover Coverage</td>
<td>1.0-SNAPSHOT</td>
<td>8</td>
<td>25.0%</td>
<td>2010-10-26</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sample Project - One Dependency</td>
<td>1.0-SNAPSHOT</td>
<td>8</td>
<td>25.0%</td>
<td>2010-10-26</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sample Project - Wicket With Unneeded Dependencies</td>
<td>1.0-SNAPSHOT</td>
<td>53</td>
<td>0.0%</td>
<td>2010-10-27</td>
<td></td>
</tr>
<tr>
<td></td>
<td>babble</td>
<td>1.0.0-SNAPSHOT</td>
<td>156</td>
<td>67.9%</td>
<td>2010-06-29</td>
<td></td>
</tr>
<tr>
<td></td>
<td>babble-core</td>
<td>1.0.0-SNAPSHOT</td>
<td>95</td>
<td>57.9%</td>
<td>2010-09-30</td>
<td></td>
</tr>
</tbody>
</table>

Powered by SonarSource - Open Source LGPL - v.2.4 - Plugins - Documentation - Ask a question - Bug/feature request
and drill-downs
<table>
<thead>
<tr>
<th>Alert</th>
<th>Name</th>
<th>Version</th>
<th>Lines of code</th>
<th>Rules compliance</th>
<th>Build date</th>
<th>Links</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Apache Log4j</td>
<td>1.2.17-SNAPSHOT</td>
<td>20,638</td>
<td>76.4%</td>
<td>2010-11-25</td>
<td></td>
</tr>
<tr>
<td></td>
<td>JUnit</td>
<td>4.8.2</td>
<td>6,419</td>
<td>67.0%</td>
<td>2010-11-25</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sample Multimodule Java Parent</td>
<td>1.0-SNAPSHOT</td>
<td>48</td>
<td>25.0%</td>
<td>2010-02-05</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sample Project - Algorithms</td>
<td>1.0-SNAPSHOT</td>
<td>8</td>
<td>25.0%</td>
<td>2010-02-05</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sample Project - Clover Coverage</td>
<td>1.0-SNAPSHOT</td>
<td>8</td>
<td>25.0%</td>
<td>2010-02-05</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sample Project - One Dependency</td>
<td>1.0-SNAPSHOT</td>
<td>8</td>
<td>25.0%</td>
<td>2010-10-26</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sample Project - Wicket With Unneeded Dependencies</td>
<td>1.0-SNAPSHOT</td>
<td>53</td>
<td>0.0%</td>
<td>2010-10-27</td>
<td></td>
</tr>
<tr>
<td></td>
<td>babble</td>
<td>1.0.0-SNAPSHOT</td>
<td>156</td>
<td>67.9%</td>
<td>2010-06-29</td>
<td></td>
</tr>
<tr>
<td></td>
<td>babble-core</td>
<td>1.0.0-SNAPSHOT</td>
<td>95</td>
<td>57.9%</td>
<td>2010-09-30</td>
<td></td>
</tr>
</tbody>
</table>
the creation of clear, understandable visualizations of data is of fundamental importance in all branches of science.

Anne E. Egger, Ph.D.
Visualizing Scientific Data:
An essential component of research
Mechanics
Install, setup, config
Download the zip

current version is 2.7
Sonar is an open source project hosted at Codehaus. Download and install your own copy.

Version: 2.4.1 (November 18, 2010), License: LGPLv3.

All in one
Sonar is an open platform to manage code quality. As such, it covers the 7 axes of code quality:
- Architecture & Design
- Comments
- Coding rules
- Duplications
- Unit tests
- Potential bugs
- Complexity

In 3 clicks
Sonar has got a very efficient way of navigating, a balance between high-level view, dashboard, TimeMachine and defect hunting tools. This enables to quickly uncover projects and/or components that are in Technical Debt to establish action plans.

Hunting
- Portfolio View
- Project Insight
- Source Code

Extend with plugins
Covering new languages, adding rules engines, computing advanced metrics can be done through a powerful extension mechanism. More than 30 plugins are already available.

Languages covered
Java is built in. Open Source and commercial plugins enable to cover Flex, PHP, PL/SQL, Cobol and Visual Basic.

Get started
1. Download
2. Unzip and start
3. Analyze projects
4. Ready to improve quality

Sonar in action
To see more, visit Nemo, the online instance of Sonar dedicated to open source projects.

Extend Sonar
- Install plugins
- Integrate to Continuous Integration servers
- Integrate to Eclipse and IntelliJ IDEA

Covered by SonarSource
SonarSource proposes commercial extensions to cover additional languages and manage portions of projects along with Professional services.
Unpack it
Put `$SONAR_HOME/bin` on your path
Type `sonar.sh start`
Database type?
✓ Derby
✓ Derby
✓ Oracle
✓ Derby
✓ Oracle
✓ MySQL
✓ Derby
✓ Oracle
✓ MySQL
✓ MS SQL
## Install Sonar

Added by Olivier Gaudin, last edited by Freddy Mallet on Nov 17, 2010 (view change)

### Table of content
- Sonar requirements
  - Server
  - Browser
  - Technical architecture
  - The 2 minutes tutorial
  - The 1 minute installation on Gentoo Linux
  - How to upgrade
    - Release Upgrade Notes
  - Full installation in 5 steps
    - Step 1 - Create database
    - Step 2 - Install server
    - Step 3 - Configure database
    - Step 4 - Start server
    - Step 5 - Configure Maven
  - Run as a service
  - Before version 2.2: declare the Sonar Maven Repository inside Nexus
  - Running Sonar behind a Proxy

## Sonar requirements

Sonar is a web application and a maven plugin using both a database. This means that most Sonar users interact with Sonar through web browsers from any computer.

### Server

Server runs on any operating system that support Java and Maven. Those two pieces of software need to be installed first:

- Java Development Kit 1.5 or later
- Maven 2.0.9+, 2.1.x, 2.2.x or 3.x (Since Sonar 2.4)
- Oracle 10c+ waterslide only

For more information and advice for the installation of Sonar, refer to the Sonar Installers Guide.

---

[http://docs.sonarsource.org/display/SONAR/Install+Sonar](http://docs.sonarsource.org/display/SONAR/Install+Sonar)
Analyzing

Measuring the code
#To forcefully get a Sonar 2.4, Maven 3 compatible plugin

mvn org.codehaus.mojo:sonar-maven-plugin:2.0-beta-1:sonar
#...Or if you’ve never used Sonar before

mvn sonar:sonar
zeus ~/Documents/Teach/Courses/Sonar/code-to-analyze/junit on master (r4.8.1-92-gd46478d) 1 commit since tracked remotes/origin/HEAD (r4.8.1-91-g81b50e9)
31 junit:master % 20:45:53
Frequency and CI

Hudson, Bamboo and Friends
How often?
Not too often
But enough to get **good granularity**
Once per day
Twice per day
(a reasonable maximum)
Why measure?
Humans, numbers, and bad intuition
managers distrust developers impulses
impulses are incongruent with...
our **personal** best interests
our **corporate** best interests
better at **team agility**
better at leveraging frameworks
as **bad** as ever choosing “what next?”
JUnit: What to fix first

1. Tightly coupled classes?
2. Empty if-statements?
3. Test coverage?
4. Magic numbers?
please don’t guess!
please don’t guess!
please don’t guess!
analysis tools can provide the data
“As the performance of the underlying systems improved... our ability to predict the performance of a program has eroded”

— Josh Bloch
Chief Java Architect at Google
measure, measure, measure
act on that measurement’s implications
measure again
tiring
who has time to glue the tools together?
automate it!
Our Shortcomings
Avoiding common mistakes
sonar
Quality is under control
“seven deadly developer sins”
“seven deadly developer sins”
Size / Complexity
Unit Tests
Duplication
Coding Rules
Potential Bugs
Architecture
Comments
Size / Complexity
Unit Tests
Duplication
Coding Rules
Potential Bugs
Architecture
Comments
Size / Complexity
Unit Tests
Duplication
Coding Rules
Potential Bugs
Architecture
Comments
Size / Complexity
Unit Tests
Duplication
Coding Rules
Potential Bugs
Architecture
Comments
Size / Complexity
Unit Tests
Duplication
Coding Rules
Potential Bugs
Architecture
Comments
Size / Complexity
Unit Tests
Duplication
Coding Rules
Potential Bugs
Architecture
Comments
Size / Complexity
Unit Tests
Duplication
Coding Rules
Potential Bugs
Architecture
Comments
The Metrics

Revealing the seven sins
complexity
• Lack of Cohesion Among Method of Class
• Number of “connected components” in a class.
  • Set of related methods and fields.
  • Should be only one such component in each class.
  • If 2 or more components, the class should be split.
Cyclomatic Complexity

- Count of:
  - if
  - for
  - while
  - case
  - catch
  - throw
  - return (not at end of method)
  - &&
  - ||
  - ?
afferent coupling

• How many classes use this one?

• “Inbound links”
efferent coupling

- How many other classes does this one use?
- “Outbound links”
size
lines of code

- One of the few raw dimensions
unit tests
Tests

• Line coverage
• Branch coverage
• % Passing
duplication
Duplication

- Duplicated code blocks
- Refactor into a single method
Private methods
Protected method
coding rules
Code styling

- Selling source code?
- For readability
potential bugs
• Rule-based
• Common mistakes
• Risky practices
architecture
- **Package tangle**
  - \(2 \times \frac{\text{package}\_\text{tangles}}{\text{package}\_\text{edges}\_\text{weight}} \times 100\)
comments
Comments

- Documented functions, classes
- Undocumented APIs
- Commented lines of code
time

the z axis
Timeline

- Uses Google charting
Motion Chart

- Dimensions not visible in numbers
API Changes

- Backwards compatible APIs
Build tools
maven 2
or
maven 3
Bootstrap it with a tiny Maven POM
<?xml version="1.0"?>
    <modelVersion>4.0.0</modelVersion>
    <groupId>[YOUR.ORGANIZATION]</groupId>
    <artifactId>[YOUR.PROJECT]</artifactId>
    <name>[YOUR PROJECT NAME]</name>
    <version>[YOUR PROJECT VERSION]</version>

    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <configuration>
                    <source>1.5</source>
                    <target>1.5</target>
                    <excludes>
                        <exclude>**/.*</exclude>
                    </excludes>
                </configuration>
            </plugin>
        </plugins>
    </build>

    <properties>
        <sonar.dynamicAnalysis>false</sonar.dynamicAnalysis>
    </properties>
</project>
Gradle
a better way to build
Bootstrap it with a tiny Maven POM
<?xml version="1.0"?>
  <modelVersion>4.0.0</modelVersion>
  <groupId>[YOUR.ORGANIZATION]</groupId>
  <artifactId>[YOUR.PROJECT]</artifactId>
  <name>[YOUR PROJECT NAME]</name>
  <version>[YOUR PROJECT VERSION]</version>

  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <configuration>
          <source>1.5</source>
          <target>1.5</target>
          <excludes>
            <exclude>**/*.*</exclude>
          </excludes>
        </configuration>
      </plugin>
    </plugins>
  </build>

  <properties>
    <sonar.dynamicAnalysis>false</sonar.dynamicAnalysis>
  </properties>
</project>
Gradle Integration with Sonar

**Details**

- **Type:** New Feature
- **Priority:** Major
- **Affects Version/s:** None
- **Component/s:** plugins
- **Status:** Open
- **Resolution:** Unresolved
- **Fix Version/s:** 1.0-milestone-2

**Description:**

It would be nice to have gradle integrate with sonar, currently there is no way this can be done. I have tried integrating gradle + hudson + sonar by means of hudson plugin configuration but that just does not work. Please see if this could be done, i would assume it could be big plus.

**Activity**

- **Gregory Boissinot** added a comment - 04/Apr/10 5:43 AM
  Sonar is deeply coupled with Maven.
  Sonar uses Maven plugins for launching metrics tools and it uses the Maven conventions for recording the generated XML files.

  It would be great whether Sonar uses Gradle for launching metrics tools and for recording the generated XML files. In addition, Gradle makes its possible to use the Maven conventions, and it’s easier.

  Therefore this issue is not a Gradle issue but a Sonar issue.
  I suggest raising a Sonar issue and I suggest closing this Gradle issue.

- **Hans Dockter** added a comment - 17/Apr/10 9:50 AM
  As Gregory pointed out, there is not a good way yet to integrate Sonar with Gradle. But it is definitely something we would like to offer at one point once Sonar is a little bit more build tool agnostic.
Jenkins?
install the Hudson Sonar Plugin
## Hudson Dashboard

### Build Executor Status

<table>
<thead>
<tr>
<th>#</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Idle</td>
</tr>
<tr>
<td>2</td>
<td>Idle</td>
</tr>
</tbody>
</table>

### Build Queue

No builds in the queue.

### Algorithms

<table>
<thead>
<tr>
<th>Job</th>
<th>Last Success</th>
<th>Last Failure</th>
<th>Last Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-GitHub-Sample</td>
<td>13 days (#13)</td>
<td>14 days (#10)</td>
<td>1 min 25 sec</td>
</tr>
<tr>
<td>GitSCMSample-FromGitHub</td>
<td>18 days (#10)</td>
<td>N/A</td>
<td>23 sec</td>
</tr>
<tr>
<td>Maven201Showcase-FromGitHub</td>
<td>1 mo 29 days (#6)</td>
<td>N/A</td>
<td>25 sec</td>
</tr>
<tr>
<td>Sample01-FromGitHub</td>
<td>18 days (#4)</td>
<td>N/A</td>
<td>6 min 55 sec</td>
</tr>
<tr>
<td>Sample17b-FromLocal</td>
<td>8 mo 9 days (#4)</td>
<td>8 mo 9 days (#1)</td>
<td>6.4 sec</td>
</tr>
<tr>
<td>Simple-Fresstyle-Echo-to-Sysout</td>
<td>9 mo 6 days (#2)</td>
<td>N/A</td>
<td>2.2 sec</td>
</tr>
<tr>
<td>Sonar-Test</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

### Icon Legend

- S: Success
- W: Warning
- L: Latest

Legend: for all, for failures, for just latest builds

setup the Hudson Sonar Plugin
Project Sonar-Test

Permalinks

- Last build (#1), 11 min ago
- Last failed build (#1), 11 min ago
- Last unsuccessful build (#1), 11 min ago
use the Hudson Sonar Plugin
## Build History

<table>
<thead>
<tr>
<th>Job</th>
<th>Last Success</th>
<th>Last Failure</th>
<th>Last Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-GitHub-Sample</td>
<td>13 days (#13)</td>
<td>14 days (#10)</td>
<td>1 min 25 sec</td>
</tr>
<tr>
<td>GitSCMSample-FromGitHub</td>
<td>18 days (#6)</td>
<td>N/A</td>
<td>23 sec</td>
</tr>
<tr>
<td>Maven201Showcase-FromGitHub</td>
<td>1 mo 29 days (#4)</td>
<td>N/A</td>
<td>25 sec</td>
</tr>
<tr>
<td>Sample01-FromGitHub</td>
<td>18 days (#4)</td>
<td>18 days (#1)</td>
<td>6 min 55 sec</td>
</tr>
<tr>
<td>Sample17b-FromLocal</td>
<td>8 mo 9 days (#4)</td>
<td>8 mo 9 days (#1)</td>
<td>6.4 sec</td>
</tr>
<tr>
<td>Simple-Freestyle-Echo-to-Sysout</td>
<td>9 mo 6 days (#2)</td>
<td>N/A</td>
<td>2.2 sec</td>
</tr>
<tr>
<td>Sonar-Test</td>
<td>N/A</td>
<td>16 min (#1)</td>
<td>2 min 8 sec</td>
</tr>
</tbody>
</table>

**Build Queue**

No builds in the queue.

**Build Executor Status**

<table>
<thead>
<tr>
<th>#</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Idle</td>
</tr>
<tr>
<td>2</td>
<td>Idle</td>
</tr>
</tbody>
</table>

**Icon:**

- S: Success
- M: Modified
- L: [Legend](#) for all, [for failures](#), [for just latest builds](#)

Other languages
The JVM and beyond
Sonar works with Java
Java

supported out of the box
But not just Java
Groovy
<?xml version="1.0"?>
  <modelVersion>4.0.0</modelVersion>
  <groupId>my.group.id</groupId>
  <artifactId>artifactId</artifactId>
  <version>1.0</version>
  <packaging>pom</packaging>
  <name>The Name of My Project</name>

  <build>
    <sourceDirectory>src/main/groovy</sourceDirectory>
  </build>

  <properties>
    <sonar.language>grvy</sonar.language>
    <sonar.dynamicAnalysis>false</sonar.dynamicAnalysis>
  </properties>
</project>
PL/SQL
worried
Flex
PHP
C#
<?xml version="1.0"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.whatever</groupId>
  <artifactId>my-solution</artifactId>
  <version>1.2.3-SNAPSHOT</version>
  <name>My solution</name>
  <packaging>sln</packaging>

  <properties>
    <!-- NOTE : the versions and parameters may be defined as properties
    Prefer this option to the plugin specific configuration as it may be accessible to several plugins
    -->
    <visual.studio.solution>MySolution.sln</visual.studio.solution>
    <visual.test.project.pattern>*.Tests;*Test</visual.test.project.pattern>
    <dotnet.tool.version>4.0</dotnet.tool.version>
    <sonar.language>cs</sonar.language>
  </properties>

  <build>
    <plugins>
      <plugin>
        <groupId>org.codehaus.sonar-plugins.dotnet</groupId>
        <artifactId>maven-dotnet-plugin</artifactId>
        <extensions>true</extensions>
      </plugin>
      <plugin>
        <groupId>org.codehaus.mojo</groupId>
        <artifactId>sonar-maven-plugin</artifactId>
        <configuration>
          <language>cs</language>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
Seriously.
Visual Basic 6
Cobol
Sonar has become a Multi-Languages Platform

By Olivier Gaudin on September 16, 2010 » tags languages, plugins

At the beginning of this year, Freddy mentioned in the Sonar roadmap for 2010 that after version 2.0 the main objective was to enable other languages on the Sonar platform through plugins. Nine months later, we have made very good progress on this subject and I wanted to take a chance to report on it.

Java is currently the only language that is built in Sonar Core. This means that when you install the platform, the support for Java is there already. All other languages get supported through plugins. All those plugins are obviously available on the forge of plugins with documentation. But here is a slightly different view, based on whether what we call “the 7 deadly sins of the developers” get covered in Sonar for the language:

<table>
<thead>
<tr>
<th>Language</th>
<th>License</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cobol</td>
<td>Commercial</td>
<td></td>
</tr>
<tr>
<td>Flex</td>
<td>LGPL v3</td>
<td></td>
</tr>
<tr>
<td>Groovy</td>
<td>LGPL v3</td>
<td></td>
</tr>
<tr>
<td>Java</td>
<td>Built-in</td>
<td></td>
</tr>
<tr>
<td>PHP</td>
<td>LGPL v3</td>
<td></td>
</tr>
<tr>
<td>PL/SQL</td>
<td>Commercial</td>
<td></td>
</tr>
<tr>
<td>VB6</td>
<td>Commercial</td>
<td></td>
</tr>
<tr>
<td>Web</td>
<td>Apache 2.0</td>
<td></td>
</tr>
<tr>
<td>.Net</td>
<td>LGPL v3</td>
<td></td>
</tr>
</tbody>
</table>

Most of those plugins are still young, have got room for improvement and the community is working on them. But they all bring already a lot of value to the project teams.

Let's now review what is on the path in the short term:

- A C plugin will be released soon
- We plan to provide a core to the C# plugin before the end of the year
- A Natural plugin will be released in October

Of course all those plugins can be installed in a single Sonar instance to centralize the quality management of an heterogeneous portfolio of applications. No need for a tool per language anymore!
Sonar has become a Multi-Languages Platform

By Olivier Gaudin on September 16, 2010

At the beginning of this year, Freddy mentioned in the Sonar roadmap for 2010 that after version 2.0 the main objective was to enable other languages on the Sonar platform through plugins. Nine months later, we have made very good progress on this subject and I wanted to take a chance to report on it.

Java is currently the only language that is built in Sonar Core. This means that when you install the platform, the support for Java is there already. All other languages get supported through plugins. All those plugins are obviously available on the forge of plugins with documentation. But here is a slightly different view, based on whether what we call “the 7 deadly sins of the developers” get covered in Sonar for the language:

<table>
<thead>
<tr>
<th>Language</th>
<th>Size &amp; Complexity</th>
<th>Unit Tests</th>
<th>Duplication</th>
<th>Coding Rules</th>
<th>Potential Bugs</th>
<th>Architecture</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cobol</td>
<td><img src="check" alt="" /></td>
<td><img src="x" alt="" /></td>
<td><img src="check" alt="" /></td>
<td><img src="check" alt="" /></td>
<td><img src="x" alt="" /></td>
<td><img src="x" alt="" /></td>
<td><img src="warning" alt="" /></td>
</tr>
<tr>
<td>(Commercial)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flex</td>
<td><img src="warning" alt="" /></td>
<td><img src="x" alt="" /></td>
<td><img src="check" alt="" /></td>
<td><img src="check" alt="" /></td>
<td><img src="x" alt="" /></td>
<td><img src="x" alt="" /></td>
<td><img src="warning" alt="" /></td>
</tr>
<tr>
<td>(LGPL v3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Groovy</td>
<td><img src="warning" alt="" /></td>
<td><img src="x" alt="" /></td>
<td><img src="check" alt="" /></td>
<td><img src="check" alt="" /></td>
<td><img src="x" alt="" /></td>
<td><img src="x" alt="" /></td>
<td><img src="warning" alt="" /></td>
</tr>
<tr>
<td>(LGPL v3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Java</td>
<td><img src="check" alt="" /></td>
<td><img src="check" alt="" /></td>
<td><img src="check" alt="" /></td>
<td><img src="check" alt="" /></td>
<td><img src="x" alt="" /></td>
<td><img src="check" alt="" /></td>
<td><img src="check" alt="" /></td>
</tr>
<tr>
<td>(Built-in)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PHP</td>
<td><img src="check" alt="" /></td>
<td><img src="check" alt="" /></td>
<td><img src="check" alt="" /></td>
<td><img src="check" alt="" /></td>
<td><img src="x" alt="" /></td>
<td><img src="x" alt="" /></td>
<td><img src="warning" alt="" /></td>
</tr>
<tr>
<td>(LGPL v3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PL/SQL</td>
<td><img src="warning" alt="" /></td>
<td><img src="x" alt="" /></td>
<td><img src="check" alt="" /></td>
<td><img src="check" alt="" /></td>
<td><img src="x" alt="" /></td>
<td><img src="x" alt="" /></td>
<td><img src="warning" alt="" /></td>
</tr>
<tr>
<td>(Commercial)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VB6</td>
<td><img src="check" alt="" /></td>
<td><img src="x" alt="" /></td>
<td><img src="check" alt="" /></td>
<td><img src="check" alt="" /></td>
<td><img src="x" alt="" /></td>
<td><img src="x" alt="" /></td>
<td><img src="warning" alt="" /></td>
</tr>
<tr>
<td>(Commercial)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Web</td>
<td><img src="check" alt="" /></td>
<td><img src="x" alt="" /></td>
<td><img src="check" alt="" /></td>
<td><img src="check" alt="" /></td>
<td><img src="x" alt="" /></td>
<td><img src="x" alt="" /></td>
<td><img src="warning" alt="" /></td>
</tr>
<tr>
<td>(Apache 2.0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.Net</td>
<td><img src="check" alt="" /></td>
<td><img src="x" alt="" /></td>
<td><img src="check" alt="" /></td>
<td><img src="check" alt="" /></td>
<td><img src="x" alt="" /></td>
<td><img src="x" alt="" /></td>
<td><img src="warning" alt="" /></td>
</tr>
<tr>
<td>(LGPL v3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Most of those plugins are still young, have got room for improvement and the community is working on them. But they all bring already a lot of value to the project teams.

Let’s now review what is on the path in the short term:

- A C plugin will be released soon
- We plan to provide a core to the C# plugin before the end of the year
- A Natural plugin will be release in october

Of course all those plugins can be installed in a single Sonar instance to centralize the quality management of an heterogeneous portfolio of applications. No need for a tool per language anymore!
Plugins
Extending the analysis
manual install of plugins

prior to v2.4
now, a **web UI** for plugins
but first, log in...
<table>
<thead>
<tr>
<th>Name</th>
<th>Version</th>
<th>Lines of code</th>
<th>Rules compliance</th>
<th>Build date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache Log4j</td>
<td>1.2.17-SNAPSHOT</td>
<td>20,638</td>
<td>76.4%</td>
<td>2010-11-25</td>
</tr>
<tr>
<td>JUnit</td>
<td>4.8.2</td>
<td>6,419</td>
<td>67.0%</td>
<td>2010-11-27</td>
</tr>
<tr>
<td>Sample Multimodule Java Parent</td>
<td>1.0-SNAPSHOT</td>
<td>48</td>
<td>25.0%</td>
<td>2010-02-05</td>
</tr>
<tr>
<td>Sample Project - Algorithms</td>
<td>1.0-SNAPSHOT</td>
<td>8</td>
<td>25.0%</td>
<td>2010-10-26</td>
</tr>
<tr>
<td>Sample Project - Clover Coverage</td>
<td>1.0-SNAPSHOT</td>
<td>8</td>
<td>25.0%</td>
<td>2010-02-05</td>
</tr>
<tr>
<td>Sample Project - One Dependency</td>
<td>1.0-SNAPSHOT</td>
<td>8</td>
<td>25.0%</td>
<td>2010-10-26</td>
</tr>
<tr>
<td>Sample Project - Wicket With Unneeded Dependencies</td>
<td>1.0-SNAPSHOT</td>
<td>53</td>
<td>0.0%</td>
<td>2010-10-27</td>
</tr>
<tr>
<td>Struts 2</td>
<td>2.2.1</td>
<td>101,316</td>
<td>74.6%</td>
<td>2010-11-27</td>
</tr>
<tr>
<td>babble</td>
<td>1.0.0-SNAPSHOT</td>
<td>156</td>
<td>67.9%</td>
<td>2010-06-29</td>
</tr>
<tr>
<td>babble-core</td>
<td>1.0.0-SNAPSHOT</td>
<td>95</td>
<td>57.9%</td>
<td>2010-09-30</td>
</tr>
</tbody>
</table>
then plugins...
## Java profiles

<table>
<thead>
<tr>
<th>Name</th>
<th>Rules</th>
<th>Alerts</th>
<th>Projects</th>
<th>Default</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>OnlyFindBugs</td>
<td>375</td>
<td>0</td>
<td>2</td>
<td></td>
<td>Set as default</td>
</tr>
<tr>
<td>Sonar way</td>
<td>116</td>
<td>0</td>
<td></td>
<td></td>
<td>Rename, Backup</td>
</tr>
<tr>
<td>Sonar way with Findbugs</td>
<td>491</td>
<td>0</td>
<td>0</td>
<td></td>
<td>Copy</td>
</tr>
<tr>
<td>Sun checks</td>
<td>61</td>
<td>0</td>
<td>0</td>
<td></td>
<td>Copy</td>
</tr>
</tbody>
</table>
plugins galore...
Sonar Plugin Library

37 Added by Freddy Mallet, last edited by Evgeny Mandrikov on Nov 25, 2010 (view change)

Additional Metrics

Artifact Size - Reports on the size of the artifact generated by projects.
Build Stability - Reports on stability of project build using Continuous Integration engine data.
Clirr - Checks Java libraries for binary and source compatibility with older releases.
Emma - An alternative to Clover and Cobertura to measure coverage by unit tests in Java.
GreenPepper - Collects and reports tests results of executable specifications provided by GreenPepper.
JaCoCo - An alternative to Clover and Cobertura to measure coverage by unit tests in Java.
JIRA Issues -Retrieves and reports the number of project issues from JIRA.
Rules Meter - Gives information on the level of activation of projects quality profiles.
SCM Activity - Collects and reports information on commits using SCM data.
Security Rules - Enables to zoom on security rules violations to keep them under control.
SonarJ - Provides architecture/quality features accompanied by metrics about cyclic dependencies and other structural aspects using SonarJ.
Taglist - Generates a report on various tags found in the code, like #todo or /FIXME tags.
Trac - Retrieves and reports the number of project issues from a Trac instance.
Useless Code - Reports on the number of lines that can be reduced in an application.

Governance

Quality Index - Calculates a global Quality Index based on coding rules, Style, Complexity and Coverage by unit tests.
SIG Maintainability Model - An implementation of the SIG MM to evaluate the maintainability of an application.
SOQL - Quality Model (Commercial) - An implementation of the SOQL Methodology, which supports the evaluation of a software application's source code in the most objective, accurate, reproducible and automated way possible.
Technical debt - Calculates the technical debt on every component of projects with a breakdown by duplications, documentation, coverage, complexity...
Total Quality - Provides an overall measure of the quality of projects linking code quality, design, architecture, and unit testing.
Views - Portfolio Management (Commercial) - Enables aggregation of projects. Projects can be grouped into applications, applications into teams, teams into departments...

IDE

Eclipse - See defects gathered by Sonar directly in Eclipse and fix them on the spot.
IntelliJ IDEA - See defects gathered by Sonar directly in IntelliJ IDEA and fix them on the spot.

Additional Languages

C - The C plugin associated to its set of rules enables to perform objective and automated C code reviews against pre-defined or home made coding best practices.
Cobol (Commercial) - Enables to perform objective and automated Cobol code reviews against pre-defined or home made coding best practices.
Flex / ActionScript - Enables analysis of ActionScript projects into Sonar.
Groovy - Enables analysis of Groovy projects into Sonar.
PHP - Enables analysis of PHP projects by handling several tools: PHP Unit, PHP Depend, PHPMD and SQL CodeSniffer.
PL/SQL (Commercial) - Enables analysis and reporting on PL/SQL projects. As an option, the plugin can extract PL/SQL code from Oracle Forms.
Visual Basic 6 (commercial) - Enables to perform objective and automated Visual Basic 6 reviews against coding best practices.
.Net - Provides support for C# projects in Sonar.
Web - Enables analysis of web files in Sonar. Current version targets JSP and JSF.

Visualization / Reporting

Motion chart - Displays projects measures using the super sexy Google Motion Chart Gadget.
PDF Report - Generates a PDF report with the results of projects analysis.
Radiator - Displays measures using a big treemap that can then be explored.
Timeline - Displays measures history using a Google Timeline Chart to replay the past.

Integration

AntHillPro - Enables to configure and launch Sonar analysis from AntHillPro.
Bamboo - Enables to configure and launch Sonar analysis from Bamboo, the Atlassian CI engine.
Build Breaker - Makes the build fail if pre-defined alert thresholds are hit.
Crowd - Enables delegation of Sonar authentication to Atlassian Crowd.
Cutoff - Exclude files from analysis based on a cutoff date threshold, to analyze the work done on an existing code base and measure the quality of new code/changes only.
Hudson - Enables to configure and launch Sonar analysis from Hudson CI engine.
LDAP - Enables the delegation of Sonar authentication to an external system. The plugin currently supports LDAP and Microsoft Active Directory.
Piwik - Submits usage of a Sonar instance to a Piwik server.
Twitter - Creates tweet, when project analysed by Sonar.
some are commercial
most are open source
motion chart...
<table>
<thead>
<tr>
<th>Alert</th>
<th>Name</th>
<th>Lines of code</th>
<th>Technical Debt ratio</th>
<th>Coverage</th>
<th>Duplicated lines (%)</th>
<th>Build date</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️</td>
<td>MasterProject</td>
<td>6,535,572</td>
<td>14.1%</td>
<td>24.5%</td>
<td>4.4%</td>
<td>02:09</td>
</tr>
<tr>
<td>✔️</td>
<td>OPS4J Pax Exam (Build POM)</td>
<td>5,390</td>
<td>12.4%</td>
<td>24.5%</td>
<td>0.7%</td>
<td>2010-11-28</td>
</tr>
<tr>
<td>✔️</td>
<td>OPS4J Pax Logging (Build POM)</td>
<td>6,902</td>
<td>13.7%</td>
<td>0.0%</td>
<td>0.2%</td>
<td>2010-09-13</td>
</tr>
<tr>
<td>✔️</td>
<td>OPS4J Pax Reflector (Build POM)</td>
<td>3,105</td>
<td>20.1%</td>
<td>0.0%</td>
<td>1.3%</td>
<td>2010-11-28</td>
</tr>
<tr>
<td>✔️</td>
<td>OPS4J Pax Scanner (Build POM)</td>
<td>3,084</td>
<td>15.4%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>2010-11-28</td>
</tr>
<tr>
<td>✔️</td>
<td>OPS4J Pax Url</td>
<td>4,797</td>
<td>14.3%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>2010-11-28</td>
</tr>
<tr>
<td>✔️</td>
<td>OPS4J Pax Web</td>
<td>10,998</td>
<td>13.9%</td>
<td>0.0%</td>
<td>1.4%</td>
<td>2010-11-28</td>
</tr>
<tr>
<td>✔️</td>
<td>ObjectLab Kit</td>
<td>2,449</td>
<td>4.3%</td>
<td>74.1%</td>
<td>0.0%</td>
<td>2010-11-28</td>
</tr>
<tr>
<td>✔️</td>
<td>OpenEJB</td>
<td>193,027</td>
<td>19.5%</td>
<td>27.7%</td>
<td>19.9%</td>
<td>2010-09-27</td>
</tr>
<tr>
<td>✔️</td>
<td>OpenFAST</td>
<td>11,480</td>
<td>11.8%</td>
<td>54.3%</td>
<td>1.7%</td>
<td>2010-11-28</td>
</tr>
<tr>
<td>✔️</td>
<td>OpenJPA Parent POM</td>
<td>208,913</td>
<td>22.8%</td>
<td>7.0%</td>
<td>3.5%</td>
<td>2010-11-28</td>
</tr>
<tr>
<td>✔️</td>
<td>OpenNMS</td>
<td>214,957</td>
<td>8.6%</td>
<td>40.0%</td>
<td>17.6%</td>
<td>2010-11-28</td>
</tr>
<tr>
<td>✔️</td>
<td>Paul Griffiths' C programming examples</td>
<td>1,895</td>
<td>11.7%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>2010-10-18</td>
</tr>
<tr>
<td>✔️</td>
<td>PicoContainer Root</td>
<td>46,199</td>
<td>7.8%</td>
<td>67.5%</td>
<td>2.3%</td>
<td>2010-11-28</td>
</tr>
<tr>
<td>✔️</td>
<td>Sigmah</td>
<td>63,203</td>
<td>11.1%</td>
<td>26.3%</td>
<td>7.2%</td>
<td>2010-11-28</td>
</tr>
<tr>
<td>✔️</td>
<td>Silverpeas Core</td>
<td>213,836</td>
<td>18.8%</td>
<td>4.0%</td>
<td>4.6%</td>
<td>2010-09-27</td>
</tr>
<tr>
<td>✔️</td>
<td>Sonar</td>
<td>43,104</td>
<td>7.0%</td>
<td>65.3%</td>
<td>0.0%</td>
<td>01:47</td>
</tr>
<tr>
<td>✔️</td>
<td>Sonar :: Development Maven Plugin</td>
<td>91</td>
<td>5.1%</td>
<td>83.6%</td>
<td>0.0%</td>
<td>02:12</td>
</tr>
<tr>
<td>✔️</td>
<td>Sonar :: Update Center</td>
<td>1,981</td>
<td>18.1%</td>
<td>30.2%</td>
<td>0.0%</td>
<td>02:25</td>
</tr>
<tr>
<td>✔️</td>
<td>Sonar IDEs</td>
<td>4,911</td>
<td>12.6%</td>
<td>23.4%</td>
<td>0.0%</td>
<td>02:20</td>
</tr>
<tr>
<td>✔️</td>
<td>Sonatype Nexus Aggregator</td>
<td>71,929</td>
<td>15.6%</td>
<td>0.0%</td>
<td>2.9%</td>
<td>2009-12-08</td>
</tr>
<tr>
<td>✔️</td>
<td>Sonatype Plugins Aggregator</td>
<td>10,888</td>
<td>11.4%</td>
<td>14.5%</td>
<td>0.8%</td>
<td>2010-10-04</td>
</tr>
<tr>
<td>✔️</td>
<td>Sonatype Spice Aggregator</td>
<td>55,636</td>
<td>13.2%</td>
<td>28.8%</td>
<td>0.0%</td>
<td>2011-02-22</td>
</tr>
<tr>
<td>✔️</td>
<td>Spring Batch</td>
<td>24,674</td>
<td>3.0%</td>
<td>88.9%</td>
<td>0.7%</td>
<td>2011-11-23</td>
</tr>
<tr>
<td>✔️</td>
<td>Spring Security</td>
<td>26,402</td>
<td>2.8%</td>
<td>68.0%</td>
<td>0.2%</td>
<td>2010-10-15</td>
</tr>
<tr>
<td>✔️</td>
<td>Spring Web Services</td>
<td>26,675</td>
<td>6.5%</td>
<td>63.7%</td>
<td>2.6%</td>
<td>2011-11-23</td>
</tr>
<tr>
<td>✔️</td>
<td>Spring IDEs</td>
<td>95,218</td>
<td>7.7%</td>
<td>12.9%</td>
<td>2.1%</td>
<td>2011-11-23</td>
</tr>
<tr>
<td>✔️</td>
<td>Squale Project</td>
<td>26,500</td>
<td>5.6%</td>
<td>72.0%</td>
<td>0.0%</td>
<td>2011-11-23</td>
</tr>
<tr>
<td>✔️</td>
<td>Struts 1.3.9</td>
<td>50,080</td>
<td>22.5%</td>
<td>14.7%</td>
<td>20.6%</td>
<td>2011-11-23</td>
</tr>
<tr>
<td>✔️</td>
<td>Struts 2</td>
<td>102,303</td>
<td>13.4%</td>
<td>33.5%</td>
<td>2.9%</td>
<td>2010-11-23</td>
</tr>
<tr>
<td>✔️</td>
<td>Swizzle</td>
<td>6,550</td>
<td>18.7%</td>
<td>35.5%</td>
<td>0.0%</td>
<td>2011-10-25</td>
</tr>
<tr>
<td>✔️</td>
<td>Swizzle</td>
<td>6,182</td>
<td>17.0%</td>
<td>46.6%</td>
<td>0.0%</td>
<td>2011-11-23</td>
</tr>
</tbody>
</table>
Google charting engine
web connectivity may be required
exposes **coding impacts** over time
IDES

Mapping metrics to the editor
But wait, there’s more!
package org.junit.experimental.theories;

import java.lang.reflect.Field;

public class Theories extends BlockJUnit4ClassRunner {
    public Theories(Class<?> klass) throws InitializationError {
        super(klass);
    }

    @Override
    protected void collectInitializationErrors(List<Throwables<? extends Throwable>> errors) {
        super.collectInitializationErrors(errors);
    }

    private void validateDataPointFields(List<Throwables<? extends Throwable>> errors) {
        Field[] fields = getTestClass().getJavaClass().getDeclaredFields();
        for (Field each : fields) {  
            if (each.getAnnotation(DataPoint.class) != null && Modifier.isStatic(each.getModifiers())) {
                errors.add(new Error("DataPoint field " + each.getName() + " must be static"));
            }
        }
    }

    @Override
    protected void validateConstructor(List<Throwables<? extends Throwable>> errors) {
        validateOnlyOneConstructor(errors);
    }

    @Override
    protected void validateTestMethods(List<Throwables<? extends Throwable>> errors) {
        for (FrameworkMethod each : computeTestMethods()) {
            if (each.getAnnotation(Theory.class) != null) {
                errors.add(new Error("Theory method " + each.getName() + " must be static"));
            }
        }
    }
}
Discover **human-to-code** interdependencies
If you can’t measure it...
can’t prove you are spending time wisely
Measure twice
Cut once
Sonar

10 Metrics for Better Builds
Credits

- Punch Card
  http://farm1.static.flickr.com/82/247968267_49cf34e1d5_o.jpg

- Monkeys
  Flickr Creative Commons

- Cups and Balls
  http://www.flickr.com/photos/laanba/4408687132/

- Chocolate Strawberries
  http://www.flickr.com/photos/raggle/3224971811/

- http://iStockPhoto.com

- http://AmbientIdeasPhotography.com