Get ready to fight your technical debt!

Tycho, Sonar and Jacoco
The goal:

Code quality!

Standards
Code Coverage
Complexity
Design
Comments
Duplications
Tests
Why is it important?

Maintain. Use. Rely.


Learn. Adapt. Replace.

And so on.
Problem?

How to analyze it?

With so many bundles
Tools are there, but...

... reports are scattered

checkstyle, pmd, findbugs, etc.
Solution?

Unify and aggregate these reports

Sonar!

The last step towards code quality analysis
Hands-on!
From a successful Tycho build
Hands-on!
Put this build on Jenkins/Hudson

<table>
<thead>
<tr>
<th>Build</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Version de Maven</td>
<td>Maven 3.0.4</td>
</tr>
<tr>
<td>POM Racine</td>
<td>pom.xml</td>
</tr>
<tr>
<td>Goals et options</td>
<td>clean install</td>
</tr>
</tbody>
</table>
Achievement #1 unlocked!
You have a reproducible build!

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Name</th>
<th>Dernier succès</th>
<th>Dernier échec</th>
<th>Dernière durée</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>recommenders from eclipse</td>
<td>4 h 40 mn (#5)</td>
<td>N/A</td>
<td>13 mn</td>
</tr>
<tr>
<td></td>
<td></td>
<td>recommenders-sonar-e3x</td>
<td>6 j 23 h (#8)</td>
<td>N/A</td>
<td>17 mn</td>
</tr>
<tr>
<td></td>
<td></td>
<td>recommenders-sonar-e42</td>
<td>6 j 23 h (#15)</td>
<td>N/A</td>
<td>21 mn</td>
</tr>
</tbody>
</table>
**Hands-on!**

**Put Sonar at work**

Install Sonar plugin on Jenkins/Hudon.

Configure it.

Configure your build.

<table>
<thead>
<tr>
<th><strong>Sonar</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td><strong>Désactiver</strong></td>
</tr>
<tr>
<td><strong>Server Public URL</strong></td>
</tr>
<tr>
<td><strong>Nom d’utilisateur BDD</strong></td>
</tr>
<tr>
<td><strong>Mot de passe BDD</strong></td>
</tr>
<tr>
<td><strong>Driver BDD</strong></td>
</tr>
</tbody>
</table>
Hands-on (ad)!

Or use Dev@cloud from Cloudbees

CloudBees

Pain free Jenkins and Sonar.
No maintenance.
Ready to use.
And more!
Achievement #2 unlocked!

Your tech debt is under control!
The missing piece

Code coverage

Is that part of code tested or not?
So what?

On the fly instrumentation

Jacoco!

The beloved jacoco agent
Jacoco?

As easy as setting an arg to the JVM

-Djavaagent=jacoco.jar=[option1=value1[,optionN=valueN]*]

Maven plugin

Sets `{tycho.testArgLine}` to `-Djavaagent=jacoco.jar...

Pro tip: Use `<appArgLine>` to set other JVM args in Tycho Surefire plugin.
<profile>
  <id>quality</id>
  <activation>
    <activeByDefault>true</activeByDefault>
  </activation>
  <properties>
    <!-- Sonar properties -->
    <sonar.core.codeCoveragePlugin>jacoco</sonar.core.codeCoveragePlugin>
    <sonar.dynamicAnalysis>reuseReports</sonar.dynamicAnalysis>
    <!-- Points to the target folder of the global parent project -->
    <sonar.jacoco.reportPath>..../target/jacoco.exec</sonar.jacoco.reportPath>
  </properties>
</profile>

<build>
  <plugins>
    <plugin>
      <groupId>org.jacoco</groupId>
      <artifactId>jacoco-maven-plugin</artifactId>
      <version>0.5.6.201201232323</version>
      <executions>
        <execution>
          <goals>
            <goal>prepare-agent</goal>
          </goals>
          <configuration>
            <append>true</append>
            <destFile>${sonar.jacoco.reportPath}</destFile>
            <includes>
              <include>org.eclipse.recommenders.*</include>
            </includes>
          </configuration>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
Achievement #3 unlocked!

You're ready to fight your tech debt!

Lines of code
30 156
419 pages
13,364 statements
377 files

Classes
459
81 packages
281 methods
137 accessors

Violations
199
Rules compliance 99.1%

Comments
3.9%
1,226 lines
7.3% doctu API
1,469 undocumented
325 commented LOCs

Duplications
3.4%
1,423 lines
87 blocks
16 files

Package coupling index
15.2%
> 15 cycles

Complexity
2.0 /method
12.4 /class
15.1 /file
Total: 5,705

Code coverage
45.5%
47.7% line coverage
36.8% branch coverage

Unit test success
100.0%
0 failures
0 errors
364 tests
15 skipped
1:43 min

Chuck Norris
Thanks

To Nicolas De loof (@ndeloof) and Cloudbees for providing us a 1st class Dev@cloud instance for this demo.
Resources

- The base project we used: http://www.eclipse.org/recommenders/
- Jenkins: https://eclipsecon2012.ci.cloudbees.com/
- Sonar: https://eclipsecon2012.sonar.cloudbees.com/
- The maven profile: https://gist.github.com/2212435

Some bugs you should follow

- Sonar@eclipse.org: http://bugs.eclipse.org/360935
- Jacoco@jenkins: https://issues.jenkins-ci.org/browse/JENKINS-10835
Questions?

Don't forget to give your feedback: