Sonar: Manage Source Code Quality

Sang Shin
www.jPassion.com
"Learn with jPassion!"

Topics

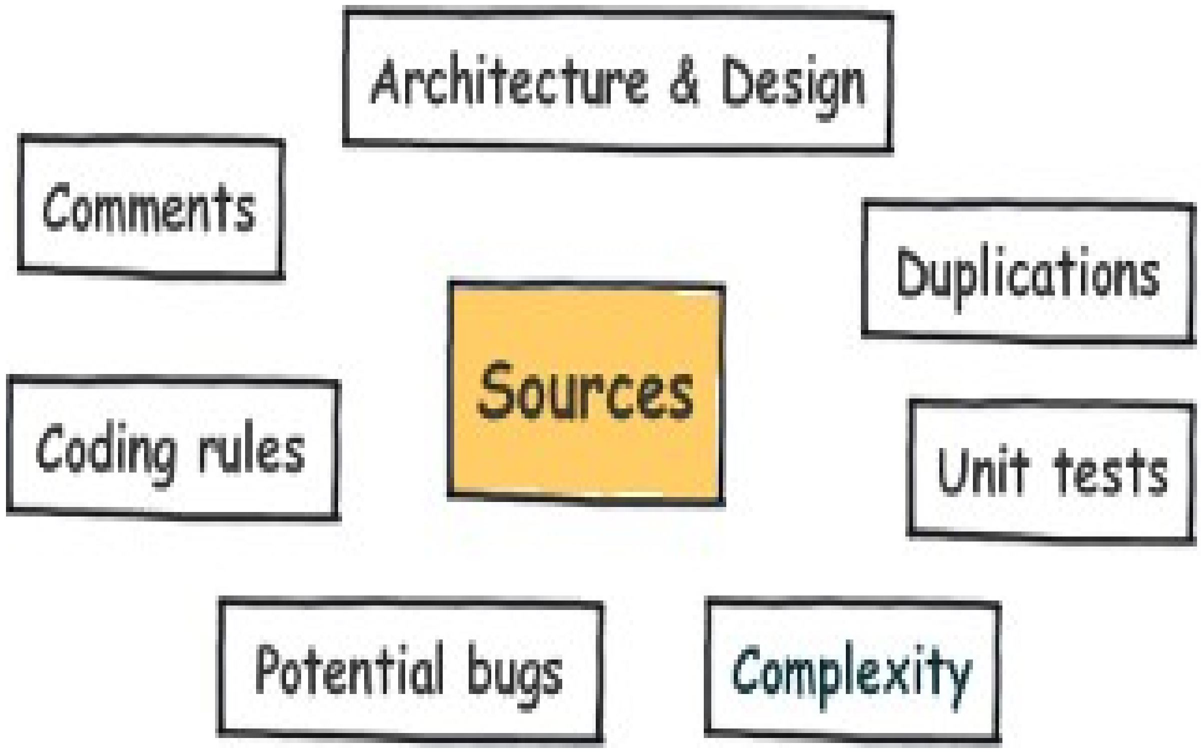
- Why manage source code quality?
- What is and why Sonar?
- Sonar architecture
- How to get started
- Integration with other tools

Why Manage Source Code Quality?

7 Deadly Sins of a Developer

- Not following coding standards and best practices
- Lacking comments in the source code, especially in public APIs
- Having duplicated lines of code
- Having complex component or/and a bad distribution of complexity amongst components
- Having no or low code coverage by unit tests, especially in complex part of the program
- Leaving potential bugs
- Having a spaghetti design

7 Axes of Software Quality



What is & Why Sonar?

What is Sonar?

- Sonar is an open source Platform used by development teams to manage source code quality
- Sonar has been developed with a main objective in mind: make code quality management accessible to everyone with minimal effort

How to Proceed on Source Code Quality Management?

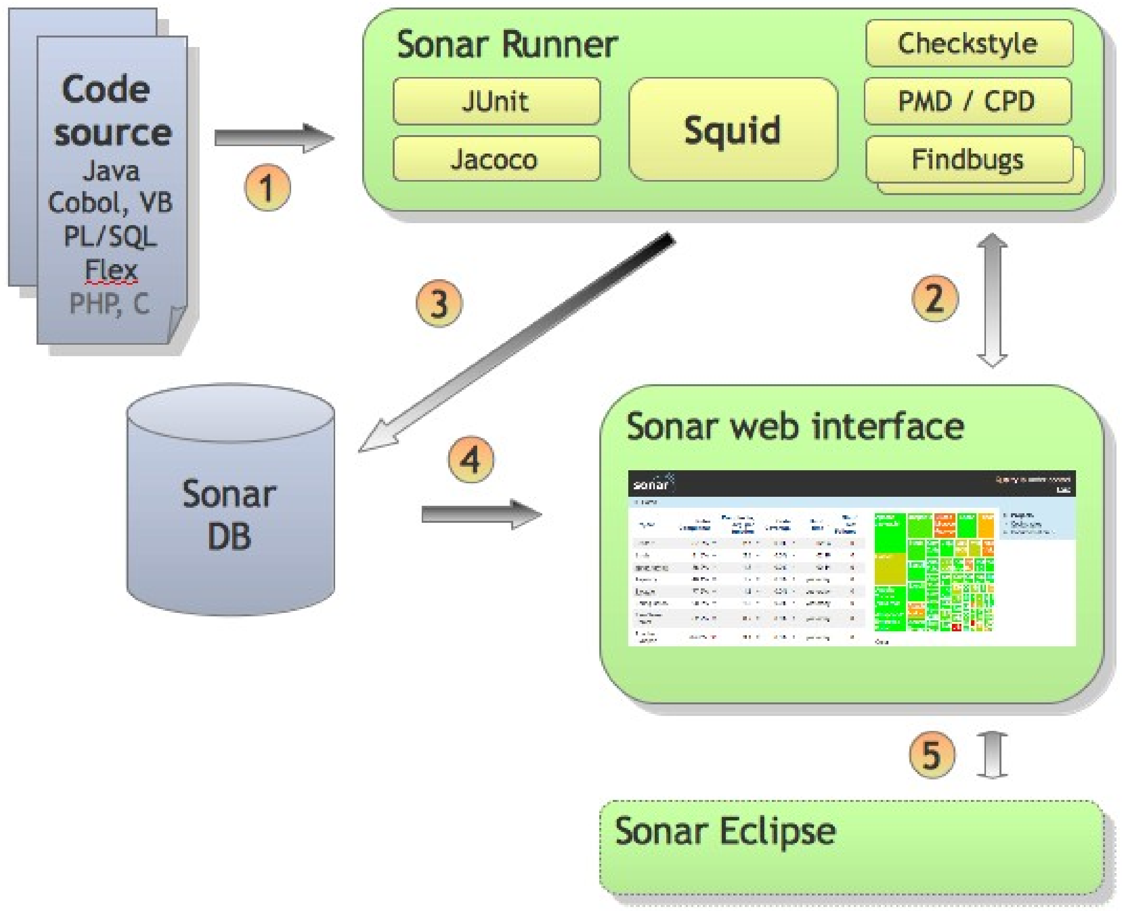
- Define which of those axes are important to you and to what extend
- Come up with a plan for reaching the target level (that might be simply to keep a high level of quality)
- Start small and go bigger when it gets fully adopted by the whole development team.

Managing Quality Profile

- Sonar enables to manage multiple quality profiles in order to adapt the required level to the type of project (only support, new project, critical application, etc).
- Managing a profile consists of:
  - > activate / deactivate / weight coding rules
  - > define thresholds on metrics for automatic alerting
  - > define project / profile association

Sonar Architecture

Sonar Technical Architecture



```
   mvn  sonar:sonar
          or
1- ant  sonar
          or
   sonar-runner
```

```
2-  http://sonar
```

Sonar Architectural components

- A set of source code analyzers
  - Grouped in a Maven plugin – Sonar can be launched through CI
  - Triggered on demand
  - Although Sonar relies on Maven to run analysis, it is capable to analyze Maven and non-Maven projects.
- A database
  - Maintains the results of the analysis, the projects and global configuration, historical analysis for TimeMachine
  - 5 database engines are currently supported : Oracle, MySQL, Derby (demo only), PostgreSQL and MS SQLServer
- A web reporting tool
  - Used to display code quality dashboards on projects, hunt for defects, check TimeMachine and to configure analysis.

Tools used by Sonar

- For finding coding rules & style violations
  - PMD
  - Checkstyle
- For finding potential bugs
  - Findbugs
- For measuring coverage by unit tests
  - Jacoco
  - Cobertura
  - Clover
- For code analyzing through source code & bytecode parsing
  - Squid

How to Get Started

Step for Getting Started

- Download the distribution from *http://sonar.codehaus.org/downloads/* and unzip it
- Open a console and start the server:
  - *$SONAR_HOME\bin\windows-x86-32\StartSonar.bat* on windows
  - *$SONAR_HOME/bin/[OS]/sonar.sh* on other platforms
- Open a console where you want to checkout the source and run
  - svn co http://svn.apache.org/viewvc/commons/proper/collections/trunk/.
- Run *mvn install sonar:sonar* in the same directory
- Browse *http://localhost:9000*

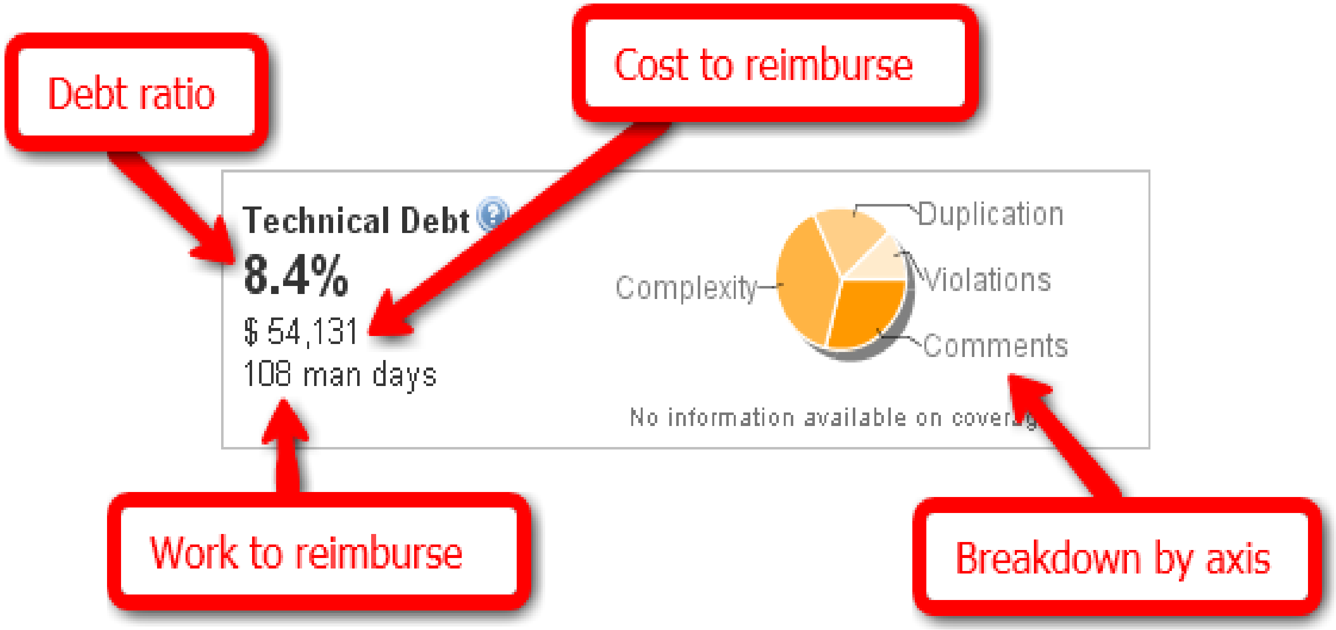
Plugin's

Plugin categories

- Additional languages
  - Flex plugin, Groovy plugin, Web plugin, XML, JavaScript plugin
- Governance
  - Technical debt plugin, Total quality plugin, etc
- Visualization & reporting
  - Radiator plugin, Motion chart plugin, Timeline plugin, Sonar PDF plugin, CSV export plugin
- Integration
  - Hudson/Jenkins plugin, Bamboo plugin
  - SCM Activity plugin, Sonar Maven report plugin, Google Analytics plugin
- Additional metrics
- Localization

Technical Debt Plugin (Page 1)

- Evaluates how much technical debt a project is in. It consists of 4 advanced measures



8.4%
$ 54,131
108 man days

No information available on coverag

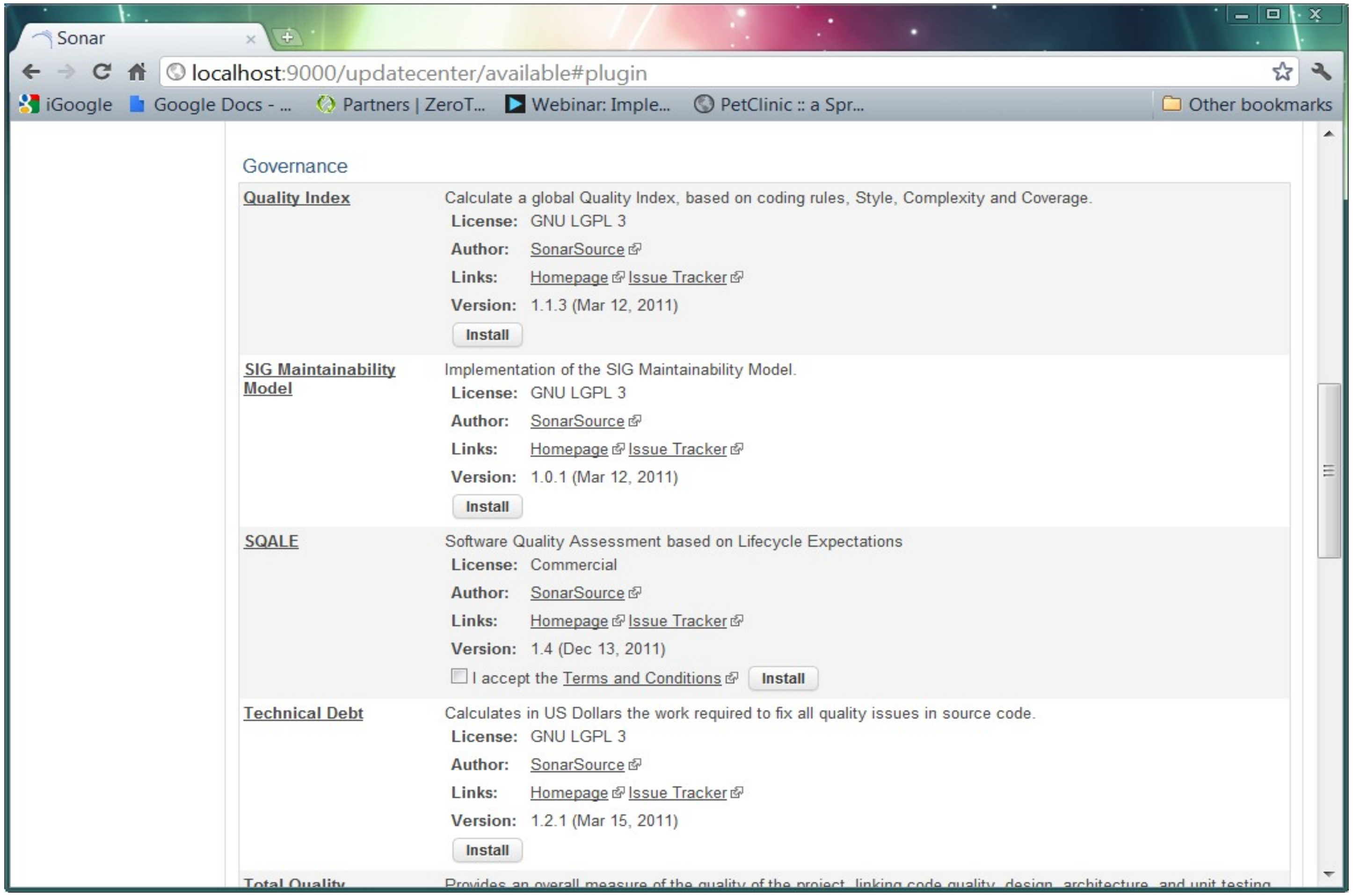
Technical Debt Plugin (Page 2)

- How it gets calculated
  - > The debt is first calculated on the basic axis : Duplication, Violations, Complexity, Coverage, Documentation and Design. It is then summed up to provide a global measure
- Explanation on measurements
  - > “debt ratio” - percentage of the current technical debt of the project versus the total possible debt for the project
  - > “cost to reimburse” - $$ what it would cost to clean all defects on every axis (no more violations, no more duplications...)
  - > “work to reimburse” - cost to reimburse expressed in man days
  - > “breakdown” - gives through a pie chart a view of the debt distribution across the 6 quality axis

Total Quality Plugin

- Combines four domains measures (architecture, design, code, and tests) in order to calculate a global and unified project quality health
  - TQ= 0.25*ARCH + 0.25*DES + 0.25*CODE + 0.25*TS
- Explanation on measurements
  - ARCH (Architecture) = 100 – TI (TI means Tangle Index)
  - DES (Design) = 0.15*NOM (Class complexity) + 0.15*LCOM (Lack of cohesion of method) + 0.25*RFC (Response for method) + 0.25*CBO (Efficient coupling) + 0.20*DIT (Depth of inheritance)
  - CODE (Code) = 0.15*DOC (Documented API density) + 0.45*RULES (Rules compliance index) + 0.40*DRYNESS (Duplicated lines density)
  - TS (Test) = Test = 0.80*COV (Code coverage) + 0.20*SUC (Unit tests success density)

Installation of Plugin's



Governance

Quality Index
Calculate a global Quality Index, based on coding rules, Style, Complexity and Coverage.
License: GNU LGPL 3
Author: SonarSource
Links: Homepage Issue Tracker
Version: 1.1.3 (Mar 12, 2011)
Install

SIG Maintainability Model
Implementation of the SIG Maintainability Model.
License: GNU LGPL 3
Author: SonarSource
Links: Homepage Issue Tracker
Version: 1.0.1 (Mar 12, 2011)
Install

SQALE
Software Quality Assessment based on Lifecycle Expectations
License: Commercial
Author: SonarSource
Links: Homepage Issue Tracker
Version: 1.4 (Dec 13, 2011)
☐ I accept the Terms and Conditions Install

Technical Debt
Calculates in US Dollars the work required to fix all quality issues in source code.
License: GNU LGPL 3
Author: SonarSource
Links: Homepage Issue Tracker
Version: 1.2.1 (Mar 15, 2011)
Install

Total Quality
Provides an overall measure of the quality of the project, linking code quality, design, architecture, and unit testing

Thank you!

Check JavaPassion.com Codecamps!
http://www.javapassion.com/codecamps
"Learn with Passion!"