Software Quality – you know it when you see it

Erik Doernenburg
ThoughtWorks
Software Quality

External perspective
• Is the software of value to its users?

Internal perspective
• How appropriate is the design?
• How easy is it to understand and extend?
• How maintainable is the software?
30,000ft and ground level

http://opensimulator.org/wiki/Grid_Architecture_Diagram

```java
public void mergePluginOutput(BuildDetail build, Map parameters,
   Iterator iterator = lines().iterator();
   while (iterator.hasNext()) {
       try {
           assemblePlugin(build, parameters, (String) iterator
       } catch (Exception e) {
           logger.error(e);
           continue;
       }
   }

   void assemblePlugin(BuildDetail build, Map parameters, String
       String className = line.trim();
       if (className.startsWith("#") || StringUtils.isEmpty(class
           return;
       }
       Class clazz = Class.forName(className);
       Widget digesterService = (Widget) clazz.newInstance();
       mergeParameters(build, parameters);
       build.addPluginOutput(digesterService.getDisplayName(), clas
   }

   private void mergeParameters(BuildDetail build, Map parameters
       parameters.put(Widget.PARAM_CC_ROOT, configuration.getCC
   }
```
The 1000ft view

Is at the right level

Aggregates data and metrics

Utilises visualisation techniques

Makes every pixel count

Depends on the question

http://softarch.97things.oreilly.com/wiki/index.php/Get_the_1000ft_view
Metrics

• lines of code
• method length
• class size
• boolean expression complexity
• cyclomatic complexity
More metrics

- duplication
- coverage
- testability
- test/code ratio

Tree maps

Show distribution of metrics

Created with checkstyle and InfoVis

http://checkstyle.sourceforge.net/4.4/
http://ivtk.sourceforge.net/
Size & complexity pyramid

Developed at Universities of Berne and Lugano
Shows key metrics and their relationships
Allows comparison to “industry standards”
Created by iPlasma tool from source code

http://loose.upt.ro/iplasma/
Code Crawler

Part of the Moose academic initiative
Allows exploration of quality metrics
Reads Famix XMI data

http://www.inf.usi.ch/faculty/lanza/codecrawler.html
http://sourceforge.net/projects/java2cdif/
CodeCity

The next step after Code Crawler

Reads Famix MSE data

http://www.inf.usi.ch/phd/wettel/codecity-download.html
Toxicity chart

Developed by ThoughtWorks
Provides easy to compare overview of quality
Created with checkstyle and Excel

Test to code ratio

Lines of unit test code per line of production code

Shows the test to code ratio over time

Created with Unix tools and Excel
Class metrics chart

Takes class as the basic entity and explore highly dimensional properties/metrics

Uses Google Docs and Motion Chart gadget

DIY

1. Get metrics
   • checkstyle, text tools, etc
   • iPlasma, et al.

2. Aggregate data
   • Ruby scripts, unix tools, etc
   • VBA and pivot tables

3. Render graphics
   • Excel is a powerful graphing tool
   • Gnuplot and InfoViz are easy to use
SpringViz

Not based on metrics, shows configuration

100 lines XSLT + 60 lines Ant + GraphViz

Test coverage

Not based on metrics, shows call graph

AspectJ + GraphViz

How do you see quality?

Comparisons

- industry standards
- different revisions: trends
- different parts: outliers

Aesthetics

- symmetry
- balance/harmony
What next?

**Measure**
- tech debt
- effectiveness of training

**Guide**
- refactoring
- clean-up

**Celebrate**
Thank you

erik.doernenburg.com
www.thoughtworks.com