The Squale Model

A Practice-Based Industrial Quality Model

Karine Mordal-Manet – Françoise Balmas – Simon Denier
Stéphane Ducasse - Harald Wertz – Jannick Laval –
Fabrice Bellingard – Philippe Vaillergues

Paris 8 University – INRIA – Qualixo - France
Software Quality

- Assess risks in software development
- Determine quality level
- Check project conformance with requirements
Quality Models

- Provide guidelines and recommendations
- Classify different categories of risks
- Referent Model:
  - Top-down models with 3 levels:
    - ISO 9126
    - MacCall Model
The Squale Model
Measures

• A measure = a raw indicator

• Automatically computable measures:
  – Metrics
  – Rules checking
  – Test coverage

• Manual measures:
  – Human expertise and audit
  – Documentation
Practices

• Introduce a new level between metrics and criteria
• Provide guidelines for developers
• Express rules to achieve optimum software quality
• Combine measures to assess conformance to rules
Criteria

- Assess quality standard:
  - Code Homogeneity
  - Comprehension
  - Simplicity
  - Interdependance level

- Designed for managers at a detailed level
Factors

• Provide an overview of the overall software quality

• 6 factors refined from Iso 9126
  – Conformity
  – Architecture
  – Maintenability
  – Evolutionarity
  – Reuse Capacity
  – Reliability
Practices in Detail: practice marks

- Weighted averages of measures

- **Individual Mark**
  - For each component targeted by the practice
  - In the range \([0;3]\)
  - Discrete or Continuous

- **Global Mark**
  - A weighted average of individual marks
  - Highlight critical practices
  - Allow to stress bad components
Practices in detail: an example
Factors dashboard

### SUMMARY OF PROJECT RESULTS

<table>
<thead>
<tr>
<th>Factor</th>
<th>Score</th>
<th>Trend</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Conformity</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Evolutionarity</td>
<td>1.8</td>
<td>🔄</td>
</tr>
<tr>
<td>Maintainability</td>
<td>2.1</td>
<td>🔄</td>
</tr>
<tr>
<td>Reliability</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Reusecapacity</td>
<td>1.9</td>
<td>🔄</td>
</tr>
</tbody>
</table>

Total: 6
## Practices dashboard

<table>
<thead>
<tr>
<th>Practice</th>
<th>Previous score</th>
<th>Score</th>
<th>Trend</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer respect</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>Dependency cycle</td>
<td>0.2</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>Stability and abstractness level</td>
<td>0.5</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>Copy paste</td>
<td>0.5</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>Class cohesion</td>
<td>1.1</td>
<td>1.1</td>
<td></td>
</tr>
<tr>
<td>Documentation standard</td>
<td>1.2</td>
<td>1.2</td>
<td></td>
</tr>
<tr>
<td>Efferent coupling</td>
<td>1.4</td>
<td>1.4</td>
<td></td>
</tr>
<tr>
<td>Swiss army knife</td>
<td>1.4</td>
<td>1.4</td>
<td></td>
</tr>
<tr>
<td>Formating standard</td>
<td>1.7</td>
<td>1.7</td>
<td></td>
</tr>
<tr>
<td>Afferent coupling</td>
<td>2.1</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td>Spaghetti code</td>
<td>2.2</td>
<td>2.2</td>
<td></td>
</tr>
<tr>
<td>Naming standard</td>
<td>2.3</td>
<td>2.3</td>
<td></td>
</tr>
<tr>
<td>Number of methods</td>
<td>2.3</td>
<td>2.3</td>
<td></td>
</tr>
<tr>
<td>Programming standard</td>
<td>2.4</td>
<td>2.4</td>
<td></td>
</tr>
<tr>
<td>Inheritance depth</td>
<td>2.9</td>
<td>2.9</td>
<td></td>
</tr>
<tr>
<td>Method size</td>
<td>3.0</td>
<td>3.0</td>
<td></td>
</tr>
<tr>
<td>Tracing standard</td>
<td>3.0</td>
<td>3.0</td>
<td></td>
</tr>
</tbody>
</table>
Practice roles

• Provide meaning to measures
• Focus on bad components
• Bridge the gap between developers and managers
• Adaptable to the enterprise requirements
• Depend on paradigm and technologies
Research and Perspectives

• Use the Squale Model to
  – Automate fix plans
  – Assess the ROI for quality increase
• Perfect the squale model to be adaptable to the quality maturity of the enterprises

• http://www.squale.org

Thank you!