Quality of student code?

- How do you objectively assess the quality of the code your students write?
- How do you make it clear to them that maintainability is important?
- Is there a way to look at maintainability without sounding subjective?

Yes there is!

The SIG benchmark model for maintainability is used to assess hundreds of industry systems every year, for a wide range of companies vital to the Dutch economy.

The SIG software maintainability model is based on the ISO/IEC 25010 standard ‘System and Software Product Quality Model’.

Assessment Process

The Open-code Clinic has been performed in the past 5 years to assess the quality of student projects for both Technical University Delft and Radboud University Nijmegen. It has been used in two forms:

One-phase Assessment

- Work hard on project
- Code review
- Guest lecture

Two-phase Assessment

- Work hard on project
- Incorporate feedback
- Final evaluation
- Code review
- Send feedback
- Code review

Assessment Results

Statistics since 2010:
- Evaluated groups: 175
- Main technologies: Java, JavaScript, C#, PHP, Python, C++
- Most problems with: Unit Size, Duplication
- Most common scores: ★★★★★, ★★★★★☆
- Percentage of applied feedback: 70%

<table>
<thead>
<tr>
<th>Volume (lines of code)</th>
<th>Maintainability (stars)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student projects tend to be small, although some are forks of open-source projects</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Volume (lines of code)</th>
<th>Lines of test code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Everyone has test code! But four projects are close to the ideal 1:1 ratio</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Volume (lines of code)</th>
<th>UNH Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Students seem to struggle when deciding to split up methods.</td>
<td></td>
</tr>
</tbody>
</table>