Teaching Software Testing with Automated Feedback

James Perretta and Andrew DeOrio, University of Michigan

ASEE Annual Conference and Exposition, June 2018
How important is it for your students to learn software testing?

RUN AAAALL THE TESTS!

I HATE TESTING
How do your students feel about it?

RUN AAAALL THE TESTS!

I HATE TESTING
Motivation

• Software testing is important!
  • But little time spent teaching it. (Edwards 2003)
• Testing takes practice.
• Automated grading becoming more common in CS courses.
Software Testing!

- 41% of IT budgets spent on QA and testing. (Hannigan & Walker 2015)
- HealthCare.gov
  - Launched Oct. 1, 2013, standard Web 2.0 app
  - Many users couldn’t register, combination of high load and software issues
  - Some applications submitted with missing info

Obama addresses healthcare website glitches

Obama: No 'sugarcoating' problems with health website
Teaching Software Testing

• Process-driven approaches:
  • Test-driven development (Desai et al 2008)
    • Test early, test often
  • SPRAE: Specification, Premeditation, Repeatability, Accountability, Efficiency (Jones & Chatman 2001)
    • Systematic approach to writing tests
Automatically Grading Student Tests

• Gives students immediate feedback on their tests.
• Test quality metrics:
  • Coverage: “What percentage of source code is exercised?”
  • Whether a test suite is free of false positives
  • Mutation Testing: “How good are tests at catching real bugs?” (true positives)

Autograder

<table>
<thead>
<tr>
<th>Student Test Suites</th>
<th>Student Tests</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student List Tests</td>
<td>✔️</td>
<td>21/21</td>
</tr>
</tbody>
</table>

List Public Tests

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Passed</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compile</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>List Public Test</td>
<td>✔️</td>
<td>1/1</td>
</tr>
<tr>
<td>Student List tests on student List</td>
<td>✔️</td>
<td>1/1</td>
</tr>
</tbody>
</table>
Mutation Testing

- Introduce small error into the code. (By hand or with automated tool)
- Run test suite.
- Any test fails == mutant exposed.

- Mutant: One copy of code with bug added.
- A high-quality test suite should expose more mutants than a low-quality test suite. (Jia & Harman 2010)
Research Questions

• Does automated feedback improve students’ ability to write high-quality test cases?
• What type of feedback best encourages student learning of software testing?

Goal: Conduct an experiment to measure the effectiveness of automated feedback policies.
Methods: Course Overview

• Population: 1,556 students over two semesters of a second-semester programming course.
• 3 hrs lecture and 2 hrs lab per week.
• Lecture and lab sections synchronized, students could attend any section and learn same material.
• Both semesters in our study synchronized for content and organization.
Methods: Programming Projects

- 5 programming projects total (we used 3 in our study):
  - Implement one or more abstract data types (ADTs).
  - Writing unit tests for the ADTs.
  - A command-line program using the ADTs.
  - Students could work alone or with a partner

<table>
<thead>
<tr>
<th></th>
<th>Project 1</th>
<th>Project 2</th>
<th>Project 3</th>
<th>Project 4</th>
<th>Project 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instructor LOC</td>
<td>140</td>
<td>301</td>
<td>595</td>
<td>372</td>
<td>495</td>
</tr>
</tbody>
</table>
Methods: Programming Projects

- 5 programming projects total (we used 3 in our study):
  - Implement one or more abstract data types (ADTs).
  - Writing unit tests for the ADTs.
  - A command-line program using the ADTs.
  - Students could work alone or with a partner

<table>
<thead>
<tr>
<th></th>
<th>Project 1</th>
<th>Project 2</th>
<th>Project 3</th>
<th>Project 4</th>
<th>Project 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instructor LOC</td>
<td>140</td>
<td>301</td>
<td>595</td>
<td>372</td>
<td>495</td>
</tr>
<tr>
<td>Average Student LOC</td>
<td>165</td>
<td>388</td>
<td>857</td>
<td>378</td>
<td>533</td>
</tr>
</tbody>
</table>
Methods: Student Test Evaluation

Student tests checked for false positives

Tests with false positives thrown out

Remaining tests run against handwritten mutants

Students awarded 1 point per mutant exposed
Example: Instructor-written Mutant

// CORRECT implementation.
template <typename T>
void List<T>::push_back(const T &datum) {
    Node *np = new Node;
    if (empty()) {
        np->prev = 0;
        first = np;
    } else {
        np->prev = last;
        last->next = np;
    }
    np->next = 0;
    np->datum = datum;
    last = np;
    ++num_nodes;
}

// BUGGY implementation: Fails if list is empty.
template <typename T>
void List<T>::push_back(const T &datum) {
    Node *np = new Node;
    np->prev = last;
    last->next = np;
    np->next = 0;
    np->datum = datum;
    last = np;
    ++num_nodes;
}
Methods: Control Group

- Students enrolled in first semester.
- Same feedback on all three projects

Autograder

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Passed</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student List test validity check</td>
<td>☒️</td>
<td>0/1</td>
</tr>
</tbody>
</table>

Test case List_test_bad.cpp incorrectly exposed the correct solution as buggy
Methods: Experiment Group

- Students enrolled in second semester.
- Additional feedback on first 2 projects.
Methods: Control & Experiment Groups

Control | Experiment
---|---

**Project 3**
- False positives
- False positives

**Project 4**
- False positives
- False positives

**Project 5**
- False positives
- False positives

Same feedback
Methods: Variables

- Independent variables:
  - Test case feedback type (control and experiment groups)
  - Partnership status
  - GPA (control for this variable)
- Dependent variables:
  - Student test case quality (percentage of mutants exposed)

We used ANOVA to look for significant associations.
## Results: Significance

<table>
<thead>
<tr>
<th></th>
<th>Project 3</th>
<th></th>
<th></th>
<th>Project 4</th>
<th></th>
<th></th>
<th>Project 5</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>df</td>
<td>Sum Sq.</td>
<td>F</td>
<td>PR(&gt;F)</td>
<td>df</td>
<td>Sum Sq.</td>
<td>F</td>
<td>PR(&gt;F)</td>
<td>df</td>
</tr>
<tr>
<td>Feedback</td>
<td>1</td>
<td>2.2</td>
<td>40.95</td>
<td><strong>2.34e-10</strong></td>
<td>1</td>
<td>3.43</td>
<td>114.92</td>
<td><strong>1.64e-25</strong></td>
<td>1</td>
</tr>
<tr>
<td>Partner</td>
<td>1</td>
<td>3.03</td>
<td>56.32</td>
<td><strong>1.31e-13</strong></td>
<td>1</td>
<td>1.59</td>
<td>53.38</td>
<td><strong>5.45e-13</strong></td>
<td>1</td>
</tr>
<tr>
<td>Feedback x Partner</td>
<td>1</td>
<td>0.01</td>
<td>0.11</td>
<td><strong>7.39e-01</strong></td>
<td>1</td>
<td>0.27</td>
<td>8.97</td>
<td><strong>2.81e-03</strong></td>
<td>1</td>
</tr>
<tr>
<td>GPA</td>
<td>1</td>
<td>25.91</td>
<td>481.46</td>
<td><strong>3.19e-88</strong></td>
<td>1</td>
<td>11.76</td>
<td>394.25</td>
<td><strong>1.08e-74</strong></td>
<td>1</td>
</tr>
<tr>
<td>GPA x Feedback</td>
<td>1</td>
<td>0.02</td>
<td>0.34</td>
<td><strong>5.60e-01</strong></td>
<td>1</td>
<td>0.0</td>
<td>0.12</td>
<td><strong>7.26e-01</strong></td>
<td>1</td>
</tr>
<tr>
<td>GPA x Partner</td>
<td>1</td>
<td>0.0</td>
<td>0.0</td>
<td><strong>9.63e-01</strong></td>
<td>1</td>
<td>0.15</td>
<td>4.9</td>
<td><strong>2.71e-02</strong></td>
<td>1</td>
</tr>
<tr>
<td>GPA x Feedback x Partner</td>
<td>1</td>
<td>0.0</td>
<td>0.07</td>
<td><strong>7.87e-01</strong></td>
<td>1</td>
<td>0.07</td>
<td>2.4</td>
<td><strong>1.21e-01</strong></td>
<td>1</td>
</tr>
<tr>
<td>Residual</td>
<td>1056</td>
<td>56.83</td>
<td></td>
<td></td>
<td>1045</td>
<td>31.17</td>
<td></td>
<td></td>
<td>991</td>
</tr>
</tbody>
</table>

Significant association b/w feedback type and test quality on all 3 projects.
### Results: Significance

<table>
<thead>
<tr>
<th></th>
<th>Project 3</th>
<th></th>
<th>Project 4</th>
<th></th>
<th>Project 5</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>df</td>
<td>Sum Sq.</td>
<td>F</td>
<td>PR(&gt;F)</td>
<td>df</td>
<td>Sum Sq.</td>
</tr>
<tr>
<td>Feedback</td>
<td>1</td>
<td>2.2</td>
<td>40.95</td>
<td>2.34e-10</td>
<td>1</td>
<td>3.43</td>
</tr>
<tr>
<td>Partner</td>
<td>1</td>
<td>3.03</td>
<td>56.32</td>
<td>1.31e-13</td>
<td>1</td>
<td>1.59</td>
</tr>
<tr>
<td>Feedback x Partner</td>
<td>1</td>
<td>0.01</td>
<td>0.11</td>
<td>7.39e-01</td>
<td>1</td>
<td>0.27</td>
</tr>
<tr>
<td>GPA</td>
<td>1</td>
<td>25.91</td>
<td>481.46</td>
<td>3.19e-88</td>
<td>1</td>
<td>11.76</td>
</tr>
<tr>
<td>GPA x Feedback</td>
<td>1</td>
<td>0.02</td>
<td>0.34</td>
<td>5.60e-01</td>
<td>1</td>
<td>0.0</td>
</tr>
<tr>
<td>GPA x Partner</td>
<td>1</td>
<td>0.0</td>
<td>9.63e-01</td>
<td>4.9</td>
<td>2.71e-02</td>
<td>1</td>
</tr>
<tr>
<td>GPA x Feedback x Partner</td>
<td>1</td>
<td>0.0</td>
<td>0.07</td>
<td>7.87e-01</td>
<td>1</td>
<td>0.07</td>
</tr>
<tr>
<td>Residual</td>
<td>1056</td>
<td>56.83</td>
<td>1045</td>
<td>31.17</td>
<td>991</td>
<td>38.12</td>
</tr>
</tbody>
</table>

- Significant association b/w partnership status and test quality on all 3 projects.
- Magnitude of association comparable to that of feedback type.
### Results: Significance

<table>
<thead>
<tr>
<th>Project</th>
<th>Feedback</th>
<th>Partner</th>
<th>Feedback x Partner</th>
<th>GPA</th>
<th>GPA x Feedback</th>
<th>GPA x Partner</th>
<th>GPA x Feedback x Partner</th>
<th>Residual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>df</td>
<td>Sum Sq.</td>
<td>F</td>
<td>PR(&gt;F)</td>
<td>df</td>
<td>Sum Sq.</td>
<td>F</td>
<td>PR(&gt;F)</td>
</tr>
<tr>
<td>Feedback</td>
<td>1</td>
<td>2.2</td>
<td>40.95</td>
<td>2.34e-10</td>
<td>1</td>
<td>3.43</td>
<td>114.92</td>
<td>1.64e-25</td>
</tr>
<tr>
<td>Partner</td>
<td>1</td>
<td>3.03</td>
<td>56.32</td>
<td>1.31e-13</td>
<td>1</td>
<td>1.59</td>
<td>53.38</td>
<td>5.45e-13</td>
</tr>
<tr>
<td>Feedback x Partner</td>
<td>1</td>
<td>0.01</td>
<td>0.11</td>
<td>7.39e-01</td>
<td>1</td>
<td>0.27</td>
<td>8.97</td>
<td>2.81e-03</td>
</tr>
<tr>
<td>GPA</td>
<td>1</td>
<td>25.91</td>
<td>481.46</td>
<td>3.19e-88</td>
<td>1</td>
<td>11.76</td>
<td>394.25</td>
<td>1.08e-74</td>
</tr>
<tr>
<td>GPA x Feedback</td>
<td>1</td>
<td>0.02</td>
<td>0.34</td>
<td>5.60e-01</td>
<td>1</td>
<td>0.0</td>
<td>0.12</td>
<td>7.26e-01</td>
</tr>
<tr>
<td>GPA x Partner</td>
<td>1</td>
<td>0.0</td>
<td>0.0</td>
<td>9.63e-01</td>
<td>1</td>
<td>0.15</td>
<td>4.9</td>
<td>2.71e-02</td>
</tr>
<tr>
<td>GPA x Feedback x Partner</td>
<td>1</td>
<td>0.0</td>
<td>0.07</td>
<td>7.87e-01</td>
<td>1</td>
<td>0.07</td>
<td>2.4</td>
<td>1.21e-01</td>
</tr>
</tbody>
</table>

- **Control for GPA**
- **Significant association b/w GPA and test quality on all 3 projects.**
Results: Test Case Quality vs. Feedback Type

All 3 differences in mean are statistically significant.
Results: Test Case Quality vs. Partnership

<table>
<thead>
<tr>
<th>Project 3</th>
<th>Project 4</th>
<th>Project 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alone</td>
<td>Partner</td>
<td>Alone</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

+14%        +9%        +8%
+4 bugs     +2 bugs    +1-2 bugs

All 3 differences in mean are statistically significant.
Limitations

• Projects in our experiment may have varied in difficulty.
• Control and experiment groups came from different semesters of same course.
  • Note: Both semesters were very consistent in organization and material.
• Students chose whether to work with a partner, who their partner would be.
Conclusion

• Students who received additional feedback on their test cases wrote higher-quality test cases, even after augmented feedback was taken away.
• Students who worked with a partner consistently wrote higher-quality test cases.
• Our work can help inform CS educators in their decisions on how to evaluate student tests and what automated feedback to provide.