Beyond Code Coverage – an Approach for Test Suite Assessment and Improvement

Dávid Tengeri*, Árpád Beszédes*, Tamás Gergely*, László Vidács†*, Dávid Havas* and Tibor Gyimóthy*

*Department of Software Engineering, University of Szeged, Hungary
†MTA-Szte Research Group on Artificial Intelligence, University of Szeged, Hungary
Industrial Motivation

WebKit

Source code

Test suite
Industrial Motivation

WebKit

• Open source web browser engine
• ~ 2.2 million lines of code (mostly C++)
• More than 27 000 test cases
Industrial Motivation

WebKit

Test suite

- Challenge with the test suite:
  - Understand and maintain the test suite
  - What is its “quality”?
Motivation and Goals

- “Quality” of a (regression) test suite:
  - What is the likelihood of defect detection?
  - How efficient, modular, etc. it is?
- How can the quality be improved?
- Quality assessment of source code – problem solved (?)
- Quality assessment of test suites?

Test suite Assessment and Improvement Method (TAIME)
Determine functional units

- Functional unit
  - Code group
  - Test group
  - subset of test suite
  - features
- statements, functions
Assessment of WebKit

<table>
<thead>
<tr>
<th>Code</th>
<th>Test</th>
<th>WebKit</th>
<th>canvas</th>
<th>css</th>
<th>dom</th>
<th>editing</th>
<th>html5lib</th>
<th>http</th>
<th>js</th>
<th>svg</th>
<th>tables</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>.53</td>
<td>.56</td>
<td>.61</td>
<td>.59</td>
<td>.67</td>
<td>.67</td>
<td>.65</td>
<td>.47</td>
<td>.50</td>
<td>.72</td>
</tr>
<tr>
<td>canvas</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.45</td>
</tr>
<tr>
<td>css</td>
<td></td>
<td>.16</td>
<td>.46</td>
<td>.26</td>
<td>.24</td>
<td>.07</td>
<td>.19</td>
<td>.00</td>
<td>.30</td>
<td>.03</td>
<td></td>
</tr>
<tr>
<td>dom</td>
<td></td>
<td>.24</td>
<td>.13</td>
<td>.51</td>
<td>.33</td>
<td>.25</td>
<td>.36</td>
<td>.00</td>
<td>.32</td>
<td>.11</td>
<td>.62</td>
</tr>
<tr>
<td>editing</td>
<td></td>
<td>.33</td>
<td>.17</td>
<td>.38</td>
<td>.52</td>
<td>.34</td>
<td>.51</td>
<td>.12</td>
<td>.35</td>
<td>.08</td>
<td>.57</td>
</tr>
<tr>
<td>html5lib</td>
<td></td>
<td>.23</td>
<td>.02</td>
<td>.31</td>
<td>.38</td>
<td>.66</td>
<td>.35</td>
<td>.01</td>
<td>.31</td>
<td>.06</td>
<td>.59</td>
</tr>
<tr>
<td>http</td>
<td></td>
<td>.29</td>
<td>.12</td>
<td>.37</td>
<td>.43</td>
<td>.46</td>
<td>.52</td>
<td>.13</td>
<td>.34</td>
<td>.20</td>
<td>.63</td>
</tr>
<tr>
<td>js</td>
<td></td>
<td>.33</td>
<td>.23</td>
<td>.41</td>
<td>.42</td>
<td>.25</td>
<td>.41</td>
<td>.65</td>
<td>.39</td>
<td>.14</td>
<td>.57</td>
</tr>
<tr>
<td>svg</td>
<td></td>
<td>.33</td>
<td>.16</td>
<td>.37</td>
<td>.47</td>
<td>.51</td>
<td>.44</td>
<td>.15</td>
<td>.44</td>
<td>.11</td>
<td>.63</td>
</tr>
<tr>
<td>tables</td>
<td></td>
<td>.26</td>
<td>.01</td>
<td>.38</td>
<td>.35</td>
<td>.17</td>
<td>.21</td>
<td>.01</td>
<td>.31</td>
<td>.50</td>
<td>.56</td>
</tr>
</tbody>
</table>

Function level coverage of groups in WebKit
The TAIME approach
The TAIME approach

Measure

- How good the tests are overall in …
  - COV: … executing all parts of the software
  - PART: … being able to localize defects
  - TpP: … being relatively few of them yet effective

- How good the test groups are in …
  - SPEC: … specializing to their code group compared to other test groups
  - UNIQ: … uniquely covering their code group compared to other code
Use cases

Assessment
- Detect any issues that require further investigation
- Helps find the initial goal in the improvement phases

One-shot refactoring
- Delete, reorganize, rewrite, create test cases
- To improve test suite quality

Change-oriented test suite evolution
- Create new test cases or possibly remove test cases
- To preserve or improve test suite quality

White-box test design
- Monitor and maintain test suite quality during design
- Use specific measurements as white-box criteria
Improvement of SoDA

- Software Development Analysis Framework
  - [http://soda.sed.hu](http://soda.sed.hu)
  - Platform independent
  - Plugin based
  - TAIME support (with GUI)

- SoDA Repository
  - Benchmark programs (SIR, WebKit, GCC)
  - Set of measurement results
## Improvement of SoDA

### White-box test design

<table>
<thead>
<tr>
<th>Func. unit</th>
<th>Tests (before)</th>
<th>Tests (after)</th>
<th>Procedures</th>
<th>Statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster</td>
<td>1</td>
<td>10</td>
<td>36</td>
<td>263</td>
</tr>
<tr>
<td>data</td>
<td>86</td>
<td>89</td>
<td>213</td>
<td>1588</td>
</tr>
<tr>
<td>fl-technique</td>
<td>2</td>
<td>4</td>
<td>16</td>
<td>175</td>
</tr>
<tr>
<td>io</td>
<td>13</td>
<td>16</td>
<td>56</td>
<td>429</td>
</tr>
<tr>
<td>metric</td>
<td>3</td>
<td>18</td>
<td>60</td>
<td>549</td>
</tr>
<tr>
<td>prioritization</td>
<td>2</td>
<td>6</td>
<td>21</td>
<td>159</td>
</tr>
<tr>
<td>reader</td>
<td>4</td>
<td>13</td>
<td>35</td>
<td>431</td>
</tr>
<tr>
<td>reduction</td>
<td>0</td>
<td>8</td>
<td>33</td>
<td>414</td>
</tr>
<tr>
<td>other</td>
<td>1</td>
<td>1</td>
<td>145</td>
<td>331</td>
</tr>
<tr>
<td>SoDA</td>
<td>112</td>
<td>165</td>
<td>615</td>
<td>4339</td>
</tr>
</tbody>
</table>
Improvement of SoDA

- Improve COV metric
  - Procedure level granularity
  - Determine groups
    - Execute
    - Measure
    - Change
    - Update groups
    - $\times 7$

- Improve SPEC metric
  - Statement level granularity
  - Propagate groups
    - Execute
    - Measure
    - Change
    - Update groups
    - $\times 28$

- Statement level granularity
  - Reuse groups
    - Execute
    - Measure
    - Change
    - Update groups
    - $\times 10$
Improvement of the *cluster* unit

Improving the coverage of *cluster* unit
Improvement of the cluster unit

Improving the coverage of other units
Improvement of the *cluster* unit

Removing the usage of *cluster* unit in other units
One-shot refactoring
White-box test design
Change-oriented test suite evolution
Assessment

http://soda.sed.hu