Testing Highly Complex System of Systems: An Industrial Case

Nauman Bin Ali, Kai Petersen, Mika V. Mäntylä

<mika.mantyla@aalto.fi>
Systems of Systems = Set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities

**Examples**

- Integrated Undersea Surveillance System (IUSS) and Sound Surveillance System (SOSUS)
- Global Positioning System (GPS)
- Telecommunication network
  - Base station (communicate with phone)
  - Administration (disable phone)

**Characteristics**

- C1 Operational independence
- C2 Managerial independence
- C3 Integration of system into system of systems
- C4 SoS comprised of complex systems
- C5 System suppliers deliver systems for integration
- C6 Complete technical overview of SoS and system
Research questions and data

- RQ1: How is SoS testing done in industry?
- RQ2: What are perceived and measured challenges when testing SoS and how are they different from testing challenges of other contexts?
- RQ3: What potential solutions do practitioners see in order to address the challenges identified?
- Data1: Software development process descriptions
- Data2: Interviews of practitioners
- Data 3: Defect data and defect slippage through process phases
RQ1: Test Process of the case company
Slippage = Defects are not found where they suppose to be found

1. Time Pressure
   - No early evaluations of software quality attributes (non-functional requirements)
2. Knowledge sharing
   - Unclear responsibilities of test levels (what defects should it find)
3. Lack of guidelines for testing at various levels/tools
   - Difficulty to do basic test as methods can not be easily executed (inheritance-depth/static methods)
   - No compliance to process (e.g. not doing reviews, not enforcing coverage targets)
   - Lack of thorough analysis to create and select test cases
   - Same developers write Basic Test, Function Test and Code
4. Quality of test cases
   - Less tests are written for new features
   - Testability of requirements (too abstract, require considerable domain expertise)
   - No responsibility for test suite (person deciding which test cases to include, exclude, etc.)
Defect Slippage can help in finding process improvement targets

- We want **bold** numbers to be 100.00%
- We want normal numbers to be 0.00%
- Numbers in *italics* are anomalies

<table>
<thead>
<tr>
<th>‘Should’ to</th>
<th>Rev</th>
<th>BT</th>
<th>FT</th>
<th>ST</th>
<th>CST</th>
<th>FOA</th>
<th>Total slippage</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘Did’ detect</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rev</td>
<td>30.76</td>
<td>7.69</td>
<td>30.76</td>
<td>19.23</td>
<td>0.00</td>
<td>11.53</td>
<td>25.71</td>
</tr>
<tr>
<td>BT</td>
<td>6.15</td>
<td><strong>26.15</strong></td>
<td>32.30</td>
<td>18.46</td>
<td>4.61</td>
<td>12.30</td>
<td>62.85</td>
</tr>
<tr>
<td>FT</td>
<td>7.14</td>
<td>50.00</td>
<td>28.57</td>
<td>14.28</td>
<td></td>
<td></td>
<td>8.57</td>
</tr>
<tr>
<td>ST</td>
<td></td>
<td>88.88</td>
<td></td>
<td>11.11</td>
<td></td>
<td></td>
<td>1.42</td>
</tr>
<tr>
<td>CST</td>
<td></td>
<td></td>
<td></td>
<td>100.00</td>
<td></td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>FOA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
</tr>
</tbody>
</table>
## Defect slippage to customer

<table>
<thead>
<tr>
<th>‘Should’ to ‘Did’ detect</th>
<th>Percentage found by Customer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rev</td>
<td>15.27</td>
</tr>
<tr>
<td>BT</td>
<td>63.88</td>
</tr>
<tr>
<td>FT</td>
<td>6.94</td>
</tr>
<tr>
<td>ST</td>
<td>11.11</td>
</tr>
<tr>
<td>CST</td>
<td>1.85</td>
</tr>
<tr>
<td>FOA</td>
<td>0.00</td>
</tr>
<tr>
<td>Customer</td>
<td>0.92</td>
</tr>
</tbody>
</table>
Maintainability of the test suite

1. Non-compliance (company proprietary code is written for testing due to lack of awareness of tools)
2. Knowledge Sharing
3. Growing number of FT test cases

Maintainability

1. Lack of readability of test cases and unclear test case descriptions
2. No detection of redundant and obsolete test cases
3. Quality of test cases

C1, C2

C1, C2

No responsibility for test suite (person deciding which test cases to include, exclude, etc.)
Lack of guidelines for testing at various levels/tools
Long turnaround time for testing

C1, C2, C4

2
- Misuse of FT framework (BT test covered here, too)
- Test cases are designed in a way that the configuration to test cycle takes time
- Limitations of FT Framework
- No selection or prioritization criteria for test cases to be run
- Growing number of FT test cases

C1, C2, C3, C4, C5

1
- Difficulty to do basic test as methods cannot be easily executed (inheritance-depth/static methods)
- Inefficient in terms of time it takes to run test cases
- Inability to do load testing
- Requires a lot of shared resources (tech. infrastructure)
- Tool-lock-in as all test cases are in the tool specific language
Improvements (RQ3)

• Developer QA: **Have more of these!**
  – Code reviews
  – Basic testing

• Function test: **Fix these!**
  – Maintenance responsibility of regression test suite
  – Controlling the size of regression suite
  – Reducing the feedback time for regression suite
  – Guidelines for FT tool usage
  – Technical improvements to FT tool
Improvements (RQ3)

• Testing in all levels
  – Template to Improve the test-case quality
  – Definitions and responsibilities of test levels:

• Requirements engineering and communication
  – Improving the testability of requirements:
  – Improved interaction between teams and cross-functional requirement teams
  – Feature status tracking
Contributions

• A study that focuses on testing challenges and solutions on SoS context
• Improve strong practices with remaining improvement potential, or weak practices?
  – Contradiction between the fault slippage measure and interviews with practitioners.
• Poor maintainability and turnaround time of functional tests are linked
  – Difficult to improve turnaround time if the test code has low maintainability.
  – Low maintainability -> developers add new test cases rather than modify the existing test cases -> Longer turnaround time
• Maintenance of test code is problematic and needs to be studied