Software Testing: State of the Art and Some trends

TOCSYC Industry Day

Robert Feldt

Blekinge Inst of Tech & Chalmers University

robert.feldt@gmail.com
Main message: Testing is in great shape!

~15-20% of Software Engineering research 2014 is **Testing**

<table>
<thead>
<tr>
<th>TOTALS</th>
<th>NumPapers</th>
<th>Avg Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation/Code</td>
<td>49</td>
<td>1,5</td>
</tr>
<tr>
<td>Testing</td>
<td>35</td>
<td>2,5</td>
</tr>
<tr>
<td>Design/models/architecture</td>
<td>23</td>
<td>4</td>
</tr>
<tr>
<td>Bugs/Quality</td>
<td>22</td>
<td>4,25</td>
</tr>
<tr>
<td>Process/Project management</td>
<td>19</td>
<td>5,25</td>
</tr>
<tr>
<td>Other V&amp;V/Model checking</td>
<td>28</td>
<td>7,75</td>
</tr>
<tr>
<td>Requirements</td>
<td>11</td>
<td>8,5</td>
</tr>
<tr>
<td>Maintenance/Evolution/Reengineering</td>
<td>10</td>
<td>8,75</td>
</tr>
<tr>
<td>Soft/Human aspects</td>
<td>5</td>
<td>13</td>
</tr>
</tbody>
</table>
Main message: Testing is in great shape!

It is a **maturing field**:

- Studies large open-source or industrial systems
- More comparison & combination of techniques
- More fundamental research with empirical methods
- Applied in new domains: Web apps & Mobile apps

But still:

- Focused on new techniques and tools, and
- Lack of Testing Process and Technique Comparisons
Some testing trends

1. “Big Data” & Machine learning for Software Testing:
   Visualize & optimize your testing

2. Automating System and Acceptance Testing:
   Visual GUI Testing can be cost effective

3. Basic Questions Studied on Industrial Data Sets
   Coverage is not a good test goal

4. Combining and Comparing Testing approaches:
   Exploratory Testing is a good complement

5. Maturing techniques: Mutation testing is practical
Some testing trends

1. “Big Data” & Machine learning for Software Testing:

   **Visualise & optimize your testing**

2. Automating System and Acceptance Testing:

   Visual GUI Testing can be cost effective

3. Basic Questions Studied on Industrial Data Sets:

   **Coverage is not a good test goal**

4. Combining and Comparing Testing approaches:

   **Exploratory Testing is a good complement**

5. Maturing techniques: Mutation testing is practical
Visualise & optimize your testing

Data + Visualisation + Discussions = Insights
Visualise & optimize your testing

<table>
<thead>
<tr>
<th>Test Start Time</th>
<th>Test Case</th>
<th>System Version</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013-09-04 04:17:12</td>
<td>Login non existing user</td>
<td>1.32 - Build 3476</td>
<td>PASS</td>
</tr>
<tr>
<td>2013-09-04 04:17:12</td>
<td>Login existing user</td>
<td>1.32 - Build 3476</td>
<td>FAIL</td>
</tr>
</tbody>
</table>
Visualise & optimize your testing

Heatmaps shows “raw” data & reveals patterns
Visualise & optimize your testing

Lack of integration traces!
Testing only what is likely to fail
Testing only what is likely to fail
Testing only what is likely to fail
Testing only what is likely to fail

Better than “80/20”!
Taking it Online
Coverage is NOT a good testing goal

Is effectiveness of a test suite correlated with its size?

Does it matter which coverage measure we use?

What if we control for test suite size?

Coverage Is Not Strongly Correlated with Test Suite Effectiveness

Laura Inozemtseva and Reid Holmes
School of Computer Science
University of Waterloo
Waterloo, ON, Canada
{lminozem,rtholmes}@uwaterloo.ca

5 systems, 1.5 MLOC, 15.5 KTests

Injected
500K mutants
Coverage is NOT a good testing goal.
Coverage is NOT a good testing goal

Is effectiveness of a test suite correlated with its size?
Yes, moderate to very high correlation

Does it matter which coverage measure we use?
No, very little difference between them

What if we control for test suite size?
Correlation between coverage and effectiveness drops

Coverage can uncover untested parts of code, but Coverage is NOT a good indicator of good testing
Exploratory Testing is a good complement

Simultaneous learning, test design and test execution
Exploratory Testing

**Criticized**: Non-systematic & automation is hard

**Recent results**:

ET is effective (experiment with 32 engineers & 97 students)

- Knowledge & skills of engineer is not critical
- Finds as many faults as scripted testing (in total)
- Finds as “severe” faults
- Finds fewer false positives
- Finds important faults outside of the testing goals
How can you take part and benefit?

Talk to the universities

Share your data, problems and goals

Try new things, continue innovating

Use thesis projects to try new ideas & start collaborations
Basic ideas: Academia in 1980’s-1990’s
MS Research project in 2006 (Pex)
In commercial IDE 2015
Thank you! Questions?

robert.feldt@gmail.com

http://www.robertfeldt.net