Assessing automated software testing with consideration of real-life incidents: A case study

Timm Gross
Roadmap

- Research gap
- Research questions
- Research approach
- Questionnaire set-up
- Questionnaire discussion
Research gap

> Automated Testing: Insurance for quality
> BUT: studies show automated testing is not as widely used as expected
> => Goal: Study how testing is used in real-life software projects
> Focus on the reasoning of developers
Research Questions (TBD)

> Does the discovery of bugs push the writing of tests?
> What is the reasoning of developers when writing (or not writing) tests?
  – What?
  – How?
  – When?
> Explore the correlation of testing and
  – The nature of the bug
  – The architecture of the code
> Are there strategies to improve fault detection by unit tests in the use case context? (i.e. Mutation Testing, etc.)
Research Approach

> Bug Analysis
  – JIRA & Bitbucket Rest API
  – General statistics (i.e. time to fix, scope, etc.)
  – Presence of tests

> Guided Questionnaires

> (Provide different testing strategies addressing identified problems, i.e. mutation testing, TDD, etc.)
Questionnaire Set-Up

> 5 Developers
> Each gets a list of 5 bugs (with tests)
> Choose one & answer the questions
> Iteration on questionnaire
> Each gets a list of 5 bugs (without tests)
> Choose one & answer the questions
> Iteration on questionnaire
> Each gets a list of 5 bugs (chosen to answer open questions)
> Choose one & answer the questions
Questionnaire Discussion

- Read the questionnaire
- Explanation
- Give feedback
  - Blindness
  - Missing aspects
  - Wording/Structure
  - Implicit knowledge
- Afterwards: Anybody interested in answering? (Bribes!)
Questionnaire
Bug Description

Please look at the list of bugs provided and familiarize yourself with them. Choose the one that you thought to be the most interesting and answer the following questions.

Please provide a short description of the bug.

Why did it happen? What was the cause of error?
Questionnaire

Bug Scope

What was the impact of the bug for the organisation (i.e. number of affected users, urgency of the fix, impediments for productive workflows)?

What was the scope of the bug in the code/repository (i.e. how much code needed to change, impacts to architecture)? How did you know?
Questionnaire
Bug Categorization

Please categorize the bug. Does it fit on one of these categories?
## Questionnaire
### Bug Categories

<table>
<thead>
<tr>
<th>Category</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic/Arithmetic</td>
<td>intention by the developer was right, but there is a mistake in the code</td>
</tr>
<tr>
<td>Structure</td>
<td>control flow/exception handling/architecture</td>
</tr>
<tr>
<td>Integration</td>
<td>interaction with external applications/resources is problematic</td>
</tr>
<tr>
<td>Data</td>
<td>cause was a problem in the data, not in the application</td>
</tr>
<tr>
<td>Function/Requirements</td>
<td>cause was a missing or misunderstood requirement</td>
</tr>
<tr>
<td>Documentation</td>
<td>missing/incomplete documentation</td>
</tr>
<tr>
<td>Infrastructure</td>
<td>configuration problem (build pipeline, server, container, etc.)</td>
</tr>
<tr>
<td>Performance</td>
<td>the required performance was not reached</td>
</tr>
<tr>
<td>Presentation</td>
<td>data is not readable by consumer (GUI, validation, etc.)</td>
</tr>
</tbody>
</table>
Did you write tests during the solution of the bug?

Have there been tests before?

If tests existed, why did they not catch the bug?
Questionnaire  
Reasoning for new tests

Please provide a short description of what you tested.

How did you test it?

When did you write the tests? Why?

Please explain your criteria to decide whether or not more tests are needed? How did you decide that you do not need further testing?
How would you rate the difficulty of testing for this bug?

What obstacles needed to be overcome when writing these tests?
Questionnaire
No Tests present

Why not?
Possible?
Desired?
Necessary?
What would you have needed to write a test?
Discussion

> I am looking forward to feedback!
General Statistics

# Issues (incl. Bugs): 1400 / with Commits: 620 / With changes of tests in the commits: 323
# Bugs: 410 / with Commits: 232 / With changes of tests in the commits: 119
License

> http://creativecommons.org/licenses/by-sa/2.5/

You are free:
• to copy, distribute, display, and perform the work
• to make derivative works
• to make commercial use of the work

Under the following conditions:

**Attribution.** You must attribute the work in the manner specified by the author or licensor.

**Share Alike.** If you alter, transform, or build upon this work, you may distribute the resulting work only under a license identical to this one.

• For any reuse or distribution, you must make clear to others the license terms of this work.
• Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.