Planning rapport for final thesis at IDA

Niclas Olofsson

February 3, 2014

1 Author

Niclas Olofsson, Civilingenjörsutbildning i Datateknik

2 Working title

Automated testing in web applications

3 Problem formulation

During code refactoring or implementation of new features in software, errors often occur in existing parts. This may have a serious impact on the reliability of the system, thus jeopardizing user's confidence for the system. Automatic testing is utilized to verify the functionality of software in order to detect bugs and errors before they end up in a production environment.

The goal of this final thesis is to analyze how automatic testing can be introduced in an existing web application in a good way, and if lower- level tests can be derived from existing high-level tests automatically. We will also study how this can be applied when extending the system with new features.

The main research questions of this project are the following:

- How can a combination of low-level and high-level testing be used when testing a web application?
- How can tests be implemented when adding new functionality into existing software?
- Is it possible to automatically derive lower-level unit tests from high-level behavioral tests?

4 Initial approach

The first step is to begin a literature study, in order to learn more about testing web applications and the challenges of that area. The Ruby language and the Rails framework will be studied since that is the context that the question formulation will be tested in. We will also look at the Knockout.js framework, which is also used in the exiting system. Furthermore, literature about automatic test generation will be read.

A test implementation for automatic test generation will be written if it seems plausible to realize. This will be tried out on tests in the existing system, and different ways of testing will also be probed by implementing new functionality into the system and study different aspects of the written tests.

In order to evaluate the results, performance factors such as code coverage and test execution times will be used. Some subjective measure of the difficulty of writing new tests will also be evaluated.

5 Literature basis

The literature basis will be focused on how automated testing can be used in a web application in order to achieve good test coverage, make tests easy to write and run them sufficiently fast.

Garousi et al. [2013] presents the results of a classification and analysis of over seventy articles about web application testing. This will be a good way to get and overview of the area, and to find other relevant articles on the subject.

To get a good and recent view on how testing typically is done in Rails, the official community guide on testing Rails applications seems like a good starting-point.

Pan et al. [2013] researches ways to automatically generate tests for web applications. This will be the starting point for the part of the question formulation that covers automatic test generation.

6 Time schedule

6.1 Week planning[1]

6.1.1 Week 5

This week is focused on getting to know the company and the existing web application, to understand the basics of project planning and which role staff manning has in this context. The project planning report should be started, a draft of the problem problem formulation should be done. Meeting with customers in order to understand what the product designed during the case study should look like.

6.1.2 Week 6

The last parts of planning takes place and the planning report is completed. The literature study begins. Focus on getting a good basic knowledge of what testing is and how it can be performed in Rails. Getting better knowledge of the involved frameworks.

6.1.3 Week 7-14

Literature study combined with test implementations, trying to determine quality factors of testing and ways to automate parts of the test suite based on other parts. Write drafts of relevant parts for the final report. Planning design of the new module to be used in evaluation study.

6.1.4 Half-time supervision

A suggested preliminary date for the half-time supervision is 2014-04-03.

The following goals should be completed at this supervision:

- A test implementation of a software to derive lower-level tests from high-level tests should be done. Alternatively, parts of the report explaining why such software would be implausible to build within the scope of this project should be accomplished.
- Draft of report parts regarding web application testing theory should be completed.
- A plan for the implementation of the new system functionality should be finished.

6.1.5 Week 15-22

Study how the previous results could be applied to existing code and how it could be used when extending the code with a new module. Implement the new functionality in order to evaluate the previous results. Writing report, hand in drafts to supervisor, correct reports (iterate as needed).

[1] Week numbers refers to the ISO 8601 week number of year

6.1.6 Week 22-23

Opposition of other final thesis. Preparations for presentation.

6.1.7 Week 23-24

Final thesis presentation. Feedback corrections on report from opponent. Post-study and evaluation report.

References

Vahid Garousi, Ali Mesbah, Aysu Betin-Can, and Shabnam Mirshokraie. A systematic mapping study of web application testing. *Information & Software Technology*, 55(8):1374–1396, 2013.

Liu Pan, Miao Huaikou, Zeng Hongwei, and Cai Lizhi. An approach to test generation for web applications. *International Journal of U- & E-Service, Science & Technology*, 6(1):61 – 75, 2013. ISSN 20054246.