Bachelor Thesis Project

Title of your thesis project
- Optional subtitle

Author: Christer Hamberg
Supervisor: Johan Hagelbäck
Semester: VT 2017
Subject: Computer Science

Abstract

The report shall begin with a summary, called abstract. The abstract shall not be longer than a paragraph, and is not divided into more than one piece. It shall contain:

- A short background description to the area of your project
- A description of your research problem
- A motivation why this problem is interesting to investigate
- What you have done to answer the problem
- A short summary of your results

From reading the abstract the reader should clearly understand what the report is all about. The purpose of the abstract is to make the reader interested in continue reading the report, if it covers something that the reader wants to know more about.

Keywords: fill in some keywords for your work here. Examples: software architectures, adaptive systems, network intrusion detection, …

Preface

You can have a preface in the report if you want, but it is not necessary. In this you can write more personal reflections on your thesis project. In the preface you can also take the opportunity to thank the people who have been particularly helpful during the report writing, for example if you had any contact with a company that helped with the project, people that guided or helped you during the project, or your family and friends that supported you during the project. The preface shall not be longer than half a page.

Contents

1 Introduction

In July 1969 a man stepped on the moon for the first time. Almost 50 years later with all the technology advances available, we still struggle to run regression testing of interactive web applications in a successful and cost efficient way.

This report provides an overview of problems observed during regression testing of an interactive commercially available web application and proposes solutions on how to overcome these issues.

In this chapter you shall give an introduction to your thesis project. It shall start with a broad overview of what your project is all about. Similar to the abstract, the introduction shall make the reader interested in continue reading your report. Don't be too detailed here; there are plenty of opportunities to add details in later chapters.

1.1 Background

<Add the clarifying picture in chapter 1.1 background>

Regression testing

The IEEE definition of regression testing is *Selective retesting of a system or component to verify that modifications have not caused unintended effects and that the system or component still complies with its specified requirements.*

Testing is done in order to build confidence that the software works as intended. When a piece of code is modified, regression test is performed and as described in the IEEE definition a selected subset of test cases is executed.

Regression testing is not done in order to test that the code change functions as intended. Instead regression testing focuses on making sure that the changed code did not break any existing functionality. Due to the repetitive nature of the test activity [REF xxx] it is one of the most expensive test activities, consuming upto 80% of the testing budget [REF yyyy].

Interactive web applications

An interactive web application is in most cases logically separated in two parts. A backend that implements the functional/business logic of the application and a frontend which implements the graphical user interface (GUI) that the user uses to interact with the system.

The logical separation is needed and used for security reasons, because the frontend does not have direct access to persistent data storage.

Backend

The backend system in a web application implements the business logic. It also provides the application with the needed access to persistent data storage, often implemented using some kind of database.

Frontend

The frontend implements the GUI needed for the user to interact with the web application. A web browser is ordinarily used for the access to the application. However the web browser itself does not implement the logic of the GUI. What it does is providing the infrastructure needed for a single-page application (SPA) to run in. The SPA is the web page which contains the code needed for the GUI logic used by the user when interacting with the web application.

API

An application programming interface (API) is used for sending of commands and data between the frontend and backend.

Document Object Model

The web browser keeps track of the web elements implemented by the SPA in an internal register, document object model (DOM). Hence different components such as buttons, text fields etc. that are available in the GUI, the locations and states of them is stored in the DOM.

Selenium WebDriver

Selenium WebDriver is an open source application that provides a programmable interface to interact with the web browser. The user can control various web elements over this interface, such as clicking buttons, entering text in text fields or validate results etc. I.e. automatically perform the tasks a user would do while using the web application.

   The web browser stores the available web components in an internal register (DOM). It is the DOM that Selenium WebDriver interacts with to do the requested tasks.

After you have described your project, you shall continue with writing a background to the area your project is in. Here you describe theories necessary to understand your project and explain terms you will use in the report.
       Example: if you do a project that is about evaluating software architectures, you describe what a software architecture is, why it is important to design an architecture that suits a specific software system, methods for evaluating and comparing different architectures, etc.

1.2 Previous research

Testing is a well-researched area focusing on what to test, how to identify overlapping test cases, how to ensure that test cases are maintained and methods to keep them up to date and functioning have been addressed.

In the scope of what to test, different methods that can be applied to select the appropriate test cases to use are presented. Also different approaches to limit the number of overlapping test cases and methods for test coverage has been researched.

The research of GUI testing is also covering multiple aspects of GUI testing, however with little coverage of the using of the GUI in regression testing.

Comparative research of different tools have been outlined in xxxx, and how creation of test cases with a GUI can be made is outlined in two main tracks:
- Recording and Playback
- Separate navigation from the flow

Recording and Playback also describes different methods that can be applied to make sure that the test cases also functions when the GUI or the navigation has changed in form of implementation of self healing test execution.

Most of the research is however done on smaller applications created specifically for the area of research instead of applying the research on commercially available systems.

< New research >
Previous research has focused on methods to limit the scope of the testing in various ways. Trial used have normally been done on smaller systems created for the purpose of the research. When GUIs have been used it is mostly a desktop application and not a web application that is used.

What is new in this report is the selected approach, to look at the possibility to combine test cases. Covering a larger scope of testing per used test case. And to research the problems that arises during the execution of the test cases rather that the problem of the scope of testing or the mass of test cases.

1. Studying the problems that arises during the execution of the test case
2. The study conducted on a commercially available web
3. Measures to address issues that arises are also tested and evaluated during the study

the problem by identifying the limiting factors, which

1.3   Problem formulation

In order to regression test a web application both frontend and backend need to be considered. However as the execution using the GUI as driver is slower compared to testing the backend utilizing the API calls directly, the GUI and the backend is tested separately from each other.

The time gained by testing the parts separately also comes with a cost, more test cases in total need to be executed and maintained. End to end test coverage is smaller than it would be in case of testing them together. Testing backend and frontend separately also opens up for possible interoperability failures. As the interface as such is not tested, failures in it might not be noticed during regression testing.

By studying and addressing the factors limiting the use of the GUI in regression testing of a commercially available web solution the intention is to show that the total amount of test cases needed can be reduced if the GUI is taken into use. The test coverage is also improved and fewer test cases are executed and maintained.

1.4 Motivation

This paper studies the possibility to combine the test cases into covering testing of both frontend and backend business logic in the same test case. Upto 80% of the testing budget can be spent on regression testing. Understanding the measures that can be applied in order to decrease the number of overlapping test cases and increasing the reliability in the test results is vital to lower the costs for regression testing of business requirements in web applications.

1.5 Research Question

RQ1.	What are the top 2 issues using Selenium WebDriver that cause longer test case execution times when test are run using a browser, compared to testing the same functionality with APIs, and what can be done to minimize the time differences?
RQ2.	What are the top 2 issues using Selenium WebDriver causing flaky (unreliably) test results, and how can these be overcome?
RQ3.	What are the top 2 issues using Selenium WebDriver causing test cases to break when GUI modifications are done, and how can these be overcome?

The expected outcome of the project is knowledge about the main issues preventing the Web GUI from being used as driver in regression testing of the business requirements in a web based application.

The project also expects to propose measures to address the issues found and show the difference in the results when those measures are applied.

RQ1	Research question 1…
RQ2	Research question 2…

1.6 Scope/Limitation

The report is limited to the scope of regression testing and focuses on factors that can be considered as addressable by the user. Selenium WebDriver is the only tool used in the study, other tools are not considered.

Factors that can be considered as not addressable are optimization of the web browser, the quality of the web browser and the web application under test.

The test execution is for sure expected to be slower when tests are run via a web browser compared to using direct API calls for testing of the business logic. However a decent "good enough" execution time is expected to be achieved.

The results might not be applicable on other web applications, as this depends on the nature of the issues that are found.

1.7 Target group

The target group is Software Testers of web applications that faces similar issues and any test organization that wishes to enhance the confidence of their regression testing.

1.8   Outline

In the following chapter the method used to approach the raised questions is outlined. The method describes both how data is collected, as well as how the proposed measures are applied.

The used implementation of the FIA application is described in the Implementations chapter, followed by a listing of the results collected using this application.

The proposed measures applied to address any issue are described under Analysis and Discussion, following by a short discussion regarding the results received when utilizing these measures.

Here you outline the rest of the report. It shall contain which chapters that will follow, and what each of them is about.

2 Method

In this chapter you shall describe the scientific approach that will be used to answer your research problem.

2.1 Scientific Approach

Here you define which formal research method(s) that shall be used to answer your research question(s). Research methods are divided into quantitative and qualitative methods. Quantitative methods result in numerical data, and qualitative in non-numerical data. You can read about different research methods [here](#).

2.2 Method Description

When you have defined which research method(s) that will be used to answer your research question(s), you shall continue by describing how the research methods will be applied in your project. If you for example shall conduct an experiment you describe your *independent* and *dependent* variables, how many times the experiment will be repeated, which software tools that will be used to run the experiment, etc. If you for example shall conduct a survey you describe how participants will be selected, how the questionnaire will be designed, how it shall be distributed to the participants, how data will be analyzed, etc.

2.3 Reliability and Validity

Here you discuss the reliability and validity of your project. You can read about reliability [here](#) and about validity [here](#). Discuss if you have any reliability issues or validity threats in your project here.

2.4 Ethical Considerations

You are required to discuss any ethical considerations (if any) in your project. If you do an experiment you will most likely not have any ethical considerations, but in a survey ethical considerations can for example be how you make sure that the privacy of the people participating in the study is not violated (by for example removing names from the gathered data).

3   Implementation

It is common that you will develop something in your project. It can be a mobile app, a stand-alone application, a website, a game, etc. In this chapter you describe the software you have implemented.

In some projects you don't develop anything, for example if you do a systematic literature review. In this case you remove this chapter.

4　Results

In this chapter you show and describe your results. You shall only show the raw results without any analysis, and you shall not put any conclusions or opinions in the description of the results. Try to be as objective as possible. An example of results from an experiment comparing five sorting algorithms is shown in Table 4.1 below.

Run	Bubble	Quick	Selection	Insertion	Merge
1	17384	24	3258	3	30
2	17559	21	3386	3	27
3	17795	19	3344	4	28
4	17484	20	3417	3	28
5	17642	19	3358	3	30
Average	17572.8	20.6	3352.6	3.2	28.6

Table 4.1: Execution times for the five sorting algorithms on 100 000 random numbers between 0 and 10 000.

What you show heavily depends on the type of research and what type of data you collect. Numerical data can for example be shown in both tables and graphs. A complementary graph for the sorting algorithms example is shown in Figure 4.1. For a questionnaire you can show the frequency (how many participants that selected the same answer) of each possible answer to a question.
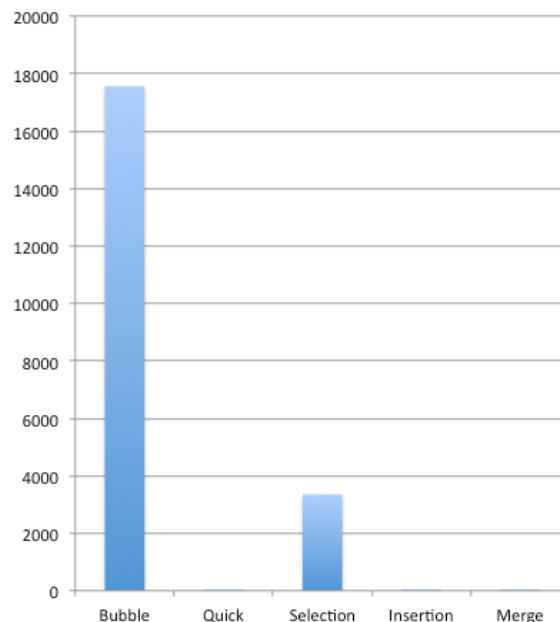

Figure 4.1: Execution times for the five sorting algorithms shown as a graph.

Note that Tables and Figures shall be labeled with chapter.number, for example Table 4.1 and Figure 1.6.

5 Analysis

Here you give meaning to and your own opinions of the results. What conclusions can you draw from the results? It is important that you don't draw any conclusions that cannot be backed up by your data. Consider using statistical tests to back up your claims. You can read about statistical testing [here](#).

6 Discussion

Here you discuss your findings and if your research question(s) have been answered. Think of research as a feedback loop. You define a problem, find a method of approaching it, execute the research and gather data. The data is then used to answer your research problem, thus creating the loop.

You shall also discuss how your findings relate to what other researchers have done in the area. Are your results similar to the findings in the related research you described in the Previous Research section?

This chapter is typically written in the present tense, while the previous chapters typically are written in past tense.

7 Conclusion

In this chapter you end your report with a conclusion of your findings. What have you shown in your project? Are your results relevant for science, industry or society? How general are your results (i.e. can they be applied to other areas/problems as well)? Also discuss if anything in your project could have been done differently to possibly get better results.

This chapter is also written in present tense.

7.1 Future Research

You cannot do everything within the limited scope of a thesis project. Here you discuss what you would do if you had continued working on your research project. Are there any open questions that you discovered during the project work that you didn't have time to investigate?

Here you shall include a list of all references used in your report. The reference list shall use the IEEE format. You can read about IEEE referencing here. In the reference list below you can find examples of how to list a webpage [1][2], a journal article [3], a book [4] and a conference proceeding (article) [5].

References

[1] Linnaeus University. (2015) Course Room for Degree Projects. [Online]. Available: https://mymoodle.lnu.se/course/view.php?id=5297#section-4

[2] Monash University. (2015, Oct. 13) Citing and referencing: IEEE. [Online]. Available: http://guides.lib.monash.edu/citing-referencing/ieee ⌷

[3] C. Lynch, "Big data: How do your data grow?" *Nature*, vol. 455, pp. 28–29, 2008.
⌷

[4] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall, 2010.
⌷

[5] D. Agrawal, S. Das, and A. E. Abbadi, "Big data and cloud computing: current state and future opportunities," in *Proceedings of the 14th International Conference on Extending Database Technology*, 2011, pp. 530–533.

[REF xxx] S. Biswas, R. Mall, M. Satpathy and S. Sukumaran Regression Test Selection Techniques: A Survey. Informatica 35 (2011), p. 289-321
<correct the missing link>

[REF yyy] P.K. Chittimalli and M.J. Harrold Recomputing coverage information to assist regression testing. IEEE Transaction of Software Engineering, 35 4 (2009), p.452-469.
<correct the missing link>

[REF zzzz] https://en.wikipedia.org/wiki/Application_programming_interface

[REF zzzz1] https://en.wikipedia.org/wiki/Single-page_application

[REF zzzz2] http://www.seleniumhq.org/

[REF zzzz3] https://www.w3.org/TR/DOM-Level-2-Core/introduction.html

[A2] Single-page application [Online]

https://en.wikipedia.org/wiki/Single-page_application

A   Appendix 1

In the appendix you can put details that does not fit into the main report. Examples are source code, long tables with raw data and questionnaires.