Lab. Report #1 – Introduction to Testing and Defect Tracking

Group #: 17

Student Names: Matteo Morrone, Adeshpal Virk, Sam Farzamfar, Taimoor Abrar

SENG 438 - Software Testing, Reliability, and Quality

Table of Contents

1 Introduction

2 High-level description of the exploratory testing plan

3 Comparison of exploratory and manual functional testing

4 Notes and discussion of the peer reviews of defect reports

5 How the pair testing was managed and team work/effort was divided

6 Difficulties encountered, challenges overcome, and lessons learned

7 Comments/feedback on the lab and lab document itself

Introduction

In this lab we were tasked with performing exploratory, manual scripted and regression testing on an ATM system. Exploratory testing is used to test the system without any planning and documentation. This type of testing is used to find major issues quickly. Manual Scripted testing is planned and uses test cases that are designed to test specific functions. This type of testing makes the test cases repeatable meaning it is easy for anyone to recreate the issue when trying to resolve a bug. Regression testing is used to discover new bugs that may have been introduced when the system was changed. This type of testing ensures that bug fixes do not introduce new issues into the system. When all three testing strategies are applied, we can ensure that all functionalities big and small are tested and work as intended.

High-level description of the exploratory testing plan

There were 7 different functions being tested in the exploratory tests in order for the testing to cover the broadest area while keeping the scope of the project in mind. These were the ON/OFF, insertion of the card, cancellations, balance inquiries, transfers,deposits, withdrawals.

For our approach we would like to test most functions a little bit as this would ensure that our system is functional most of the time. If we were to test a few functions extensively this could lead to some key functionalities having bugs which would render our system useless. We will plan tests based on following a flow that an end user would go through and any forks they would go through. This leaves out certain actions that are abnormal and might still contain bugs.

Comparison of exploratory and manual functional testing

Exploratory testing is based on a lack of planning and documentation and does not require a test script. In this lab exercise the exploratory testing consisted of manually trying various inputs while in different states of the program and observing whether the output was as we expected. Manual testing on the other hand is scripted and performs tests that have already been reviewed, with detailed steps and concrete expected results. For this lab we were given 40 of said test cases, which we performed on version 1.0 of the system for the manual functional testing requirement. There are many obvious drawbacks to exploratory testing, its lack of structure makes it difficult to say with confidence that your tests have exhausted every avenue that you otherwise could have with scripted testing, however the spontaneity of this style of testing often allows for the discovery of new bugs that otherwise would have been missed. In this lab for example, exploratory testing allowed us to discover that when in the withdraw menu, if a user selects any number on the keypad other than the listed selections of 1 through 5, the ATM displays that $20 has been withdrawn. This particular bug would not have been found through the provided manual test script alone and thus demonstrates the benefits of exploratory testing. The tradeoff, as mentioned, being that our group missed several bugs in our exploratory testing phase which were only caught through the rigorous structure of the manual test suite.

Overall, manual testing revealed more bugs than exploratory testing and therefore should be solidified as the more effective form of testing within the confines of this lab. In terms of efficiency, we found that our exploratory testing resulted in the excessive repetition of numerous steps as we did not have the foresight to know how to efficiently move through the program to maximize our testing. The beauty of manual testing is that we have the ability to organize the order of our tests to minimize time spent in between functions and eliminate unnecessary repetition of steps.

Notes and discussion of the peer reviews of defect reports

After taking a look at what each of the peer groups did it was clear that splitting the functions to each group was the smart choice as each function got tested very extensively by one group which led to much more deeper bugs to be found. Both teams found roughly 14 bugs that got condensed down to 6 after finding doubles or similar ones and grouping. These 6 bugs encompassed larger more generalized problems and were reported as such. The fact that both pairs found so many of the same bugs showed how glaring those particular issues were and made it clear that they were very likely to have been found by any regular user in the software's lifetime.

How the pair testing was managed and team work/effort was divided

We split up into two groups which were: Sam/Adesh and Matteo/Taimoor. Each pair was responsible for holding the other accountable. In each group, we decided one person would run the program and when a defect was found the other person would add it to backlog. After each bug we switched who was doing the testing and who was recording to backlog. Then after we merged the pairs back together we all went through our results as a group in order to discuss which bugs could be combined, as well as to discuss the validity of each bug.

Difficulties encountered, challenges overcome, and lessons learned

The first difficulty we encountered was collecting the necessary information when we came across any bug. A few of the bugs were hard to retrace and we had trouble knowing which functions were being tested. These issues were resolved by collectively problem solving as a group. Another difficulty we had was figuring out where we should store information within backlog itself.

Comments/feedback on the lab and lab document itself

Upon completion of this lab we noticed that the provided manual test suite was very clear and well worded, making it easy to execute the required tests with minimal confusion. The lab itself was very interesting as it gave us a glimpse into new kinds of testing that we could potentially be using in our careers in the future, as well as an introduction to new tools for said testing such as backlog. The instructions were clear and concise, and there was a helpful introduction preceding the document explaining useful terms and concepts.