**Chapter 5: Application Testing Phase**

The testing phase is one the most important and expected one, because the invented solution, that a lot of time were invested in, get to be tested. The testing phase usually runs smoothly when all other phases were implemented adequately. The testing phase is used to ensure that the solution fits its needs. In organizational level, projects may take a few different roles within the business to be played during the testing phase such as Project Sponsor, Project Manager, Testing Lead. Functional Analyst and Subject Matter Experts (SMEs) and the Technical Staff (Niesse, 2018).

When testing a software, it is important to follow a few principles for software testing in agile method as follows:

* Testing is continuous;
* Continuous feedback;
* Tests performed by the whole team;
* Decrease time of feedback response;
* Simplified & clean code;
* Less documentation;
* Test driven.

There are a few advantages of Agile Testing such as Saving time and money, Reduction of documentation, Flexible and highly adaptable to changes, provides a way for receiving regular feedback from the end user and Better determination of issues through daily meetings (reqtest, 2018).

There are several different types of testing and this chapter is focused on explaining a few of them and determine a chosen option for the proposed application.

**5.1 White-Box Testing**

White-Box testing also called Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code-Based Testing or Structural Testing is a software testing method that tests the internal structure, the design and implementation of the solution to be tested. Inputs are chosen by the tester to verify paths throughout the code and to determine the appropriate outputs (STFa, 2020).

In order for this testing to be done, the tester must have knowledge about the internal structure, the code and the program of the software. It tests the logic of the software and it is generally applicable to lower levels of software testing (Jain, 2020).

Advantages:

* It can start at an early stage, which means that there is no need to wait for the GUI to be available;
* This procedure is more thorough and has the possibility to cover most paths.

Disadvantages:

* High skilled resources with thorough knowledge of programming and implementation required once this testing can be very complex;
* It can become a burden if the changes are too frequent;
* It is expensive

(STFa, 2020)

**5.2 Black-Box Testing**

Black Box testing is also called Behavioural testing, opaque-box, closed-box, specification-based or eye-to-eye testing (Software Testing Help, 2020) in which the testing of the structure, the design and the implementation of the software being teste if not known to the tester. It is usually functional but can also be non-functional. The errors are found in this type of testing within a few categories such as Incorrect of missing functions, Interface errors, Errors in data structures or external database access, Behaviour or performance errors. The name of this testing technique is given because the software program acts like a black box in the eyes of the tester. Figure 5.2a show the structure of the Black-Box Testing (STFb, 2020).

Unlike the White-Box testing, the Black-Box Testing analyses the functionality of an application according to its specifications and tests the behaviour of the software. It is appropriate to the higher levels of testing of software and can be done by trial and errors (Jain, 2020).

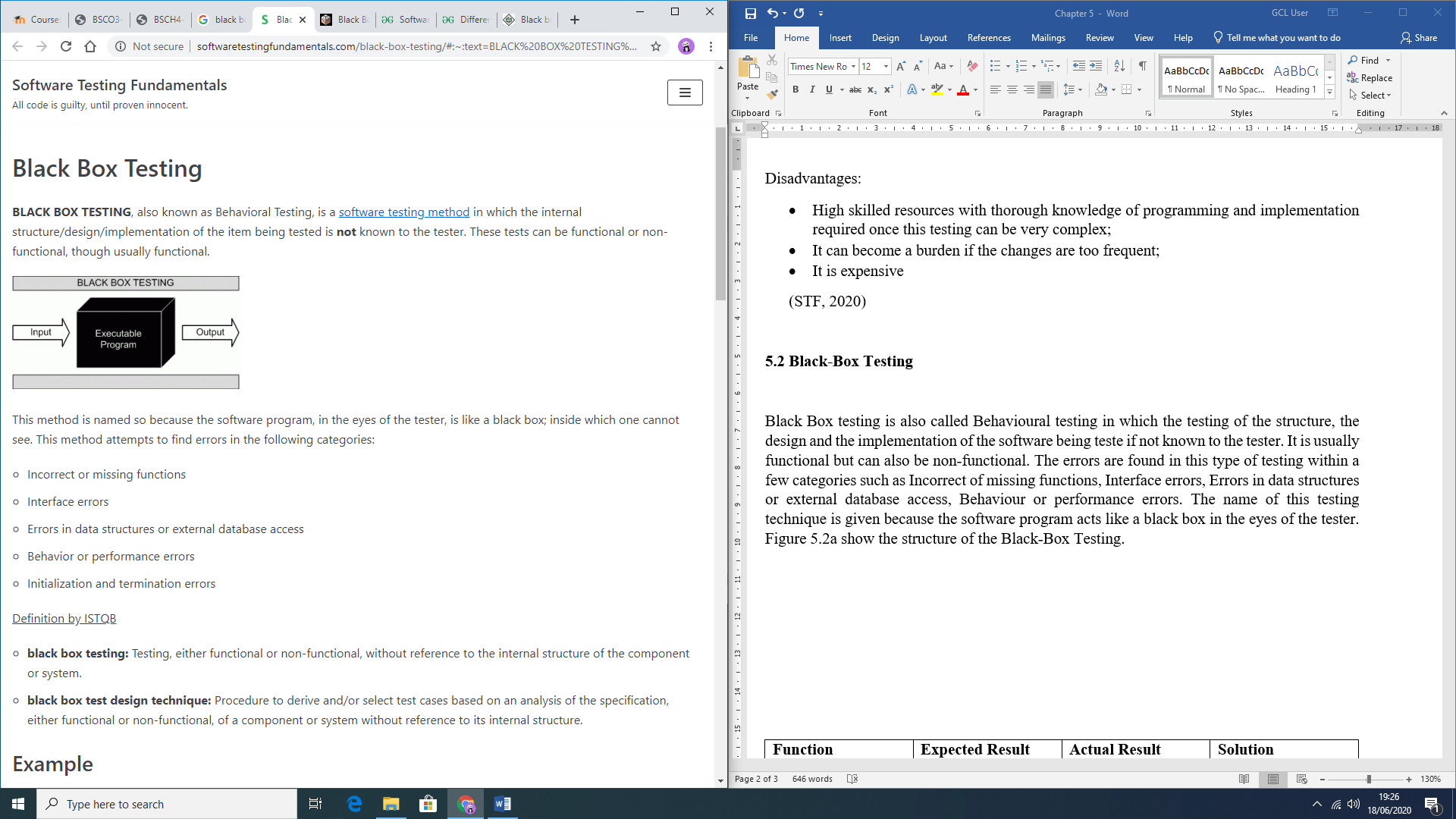
  
Advantages:

Figure 5.2a: Black-Box representation (STFb, 2020)

* The test is done using the user point of view helping to expose discrepancies in the specifications;
* There is no need for programming languages or knowledge of software;
* It can avoid developer-bias since the testing can be conducted by a body independent from the developers;
* Allows for an objective perspective;
* Once the specifications are completed, the test can be designed.

Disadvantages:

* Possibility of leaving many program paths untested since it allows for just a small number of possible inputs to be tested;
* It can be difficult to design if there are no clear specifications;
* Redundancy if the developer has already run a test case;

(STFb, 2020)

**5.3 Grey-Box Testing**

Grey-Box Testing combines both Black-Box and White-Box Testing methods. Initially, the internal structure of the software is known which means involving having access to internal data structures and algorithms in order to design the test cases, but testing at the user level (Black-Box level). The name is given because the program is like a grey or semi-transparent box in the eyes of the tester, which means that internally can be partially seen (STFc, 2020).

By being able to identify context-specific errors that belong to web systems, if the tester encounters any defect, he is able to make the changes in the code and test it again in real time. It gives the opportunity to test either the presentation layer and the internal coding structure of the software (JavaTPoint, 2020).

Advantages:

* The goals are clear for users and developers while testing;
* User focused, since the testing is mostly done by the user perspective;
* No need for high programming knowledge for this testing;
* Non-intrusive;
* The overall quality of the software is improved;
* More time to defect fixing by the developers;
* Combined benefits from Black-Box and White-box testing;
* Unbiased, avoiding conflicts between a tester and a developer;
* More effective in integration testing.

Disadvantages:

* If testing is performed for distributed systems, the defect association is difficult;
* Limited access for code path traversal lead by limited access to internal structure;
* Not possible to do a complete white-box testing because the source code cannot be accessed
* Not suitable for algorithm testing;
* Difficulty to design most of test cases.

(GeeksforGeeks, 2020)

**5.4 Chosen Testing Method**

Grey-Box testing was chosen for this project due to its combined advantages and to the fact that allows the testing of the user perspective along with the developer perspective. Errors are meant to be encountered while developing the software, however, with Grey-Box testing these errors can be fixed along with the coding. Also, the testing coverage is increased within all areas of the application and suits the limited project time frame.

**5.5 Errors Encountered During Application Development**

A few errors were encountered during the development of the software. Table 5.5a presents these errors and how they were fixed.

|  |  |  |  |
| --- | --- | --- | --- |
| Function | Expected Result | Actual Result | Solution |
| Register | Error message displayed when user presses the register button without complying with security requirements | User registered successfully even without complying with security requirements | Use of the appropriate patterns (phone, email and password) and use of appropriate validation methods |
| Use of the filters | Necessary to use one filter to activate the next and so forward | User was able to select different filters without dependency | Nest each one of the filters within the other |
| Shelter profile display | Show the appropriate information on the specific shelter | Mismatched information displayed on the screen | Passing the specific information about the shelter through an extra (putExtra()) in an intent |
| Return to Animal profile from Shelter profile | Returning to the animal profile | Error occurred leading the user to the main page | Passing the animal object through an intent when opening the shelter profile screen |

*Table 5.5a: Initial errors encountered during development and the solutions*

**5.6 Application Testing Process with Test Cases**

All functionalities were completely tested and Table 5.6a presents the checks done with Testing Cases and outcome of these tests.

Table 5.6a: Application Testing process with Test Cases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Expected result** | **Pass** | **Fail** | **Actual result** |
| Open  application | Initial screen opens if user has not registered and logged in, otherwise main screen opens |  |  | Initial screen is displayed if user has not registered and logged in. Main screen is displayed otherwise |
| Open register screen | Register screen opens |  |  | Register screen is displayed |
| Register button pressed without any data entered | Display error message to enter data for each field |  |  | Error message is displayed in each field asking user to enter data |
| Register button pressed when invalid data is entered | Display specific error message to each field where data is invalid, guiding the user on how to fix it |  |  | Specific error message is displayed to each field where data is invalid, guiding the user on how to fix it |
| Register button pressed with valid data entered in all fields | Create new user and open main screen |  |  | New user is created, and main screen is displayed |
| Open login screen | Login screen opens |  |  | Login screen is displayed |
| Login button pressed without any data | Display error message to enter data |  |  | Error message is displayed in each field asking user to enter data |
| Login button pressed when invalid data is entered | Display specific Toast message regarding the error |  |  | Specific Toast message regarding the error is displayed |
| Login button pressed with correct username and password | User is logged in successfully while a Toast messaged is displayed and main screen opens |  |  | Login is successful while a Toast message is displayed, and main screen is displayed |
| *“Not registered yet?”* TextView is clicked | Register screen opens |  |  | Register screen is displayed |
| *“Forgot your password?”* TextView is clicked | Reset password screen opens |  |  | Reset password screen is displayed |
| Reset password button is clicked without any e-mail entered | Display error message to enter username |  |  | Error message is displayed asking user to enter their username |
| Reset password button is clicked with invalid e-mail entered | Display specific error message, guiding user on how to fix it |  |  | Specific error message is displayed, guiding user on how to fix it |
| Reset password button is clicked with correct e-mail entered | An e-mail with a link for resetting the password is sent to the user’s registered e-mail |  |  | Send a resetting password link within an e-mail to the user’s registered e-mail |
| Update profile menu item is clicked | Open update profile screen |  |  | Update profile screen is displayed |
| Update profile button clicked with empty fields | Display error message to enter data for each field |  |  | Error message is displayed in each field asking user to enter data |
| Update profile button clicked when invalid data is entered | Display specific error message to each field where data is invalid, guiding the user on how to fix it |  |  | Specific error message is displayed to each field where data is invalid, guiding the user on how to fix it |
| Update profile button clicked when valid data is entered | Display Toast message notifying the user that the changes were saved |  |  | Toast message notifying the user that the changes were saved is displayed |
| Logout menu item is clicked | User is logged out while a Toast message is being displayed on the screen, and initial screen opens |  |  | Logout is successful while a Toast message is displayed, and initial screen is displayed |
| Category filter is clicked | Options of animal category are displayed in a dropdown menu |  |  | Displays the animal category options available in a dropdown menu format |
| Category option item is clicked | List of animals displayed on the screen changes to match the selected category. Size filter selection is enabled |  |  | Displays the list of animals that match the selected category and enables the size filter selection |
| Size filter is clicked | Options of size are displayed in a dropdown menu |  |  | Displays the size options in a dropdown menu format |
| Size option item is clicked | List of animals displayed on the screen changes to match the selected size while also matching the pre-selected category filter. Breed filter selection is enabled |  |  | Displays the list of animals that match the selected size while also matching the pre-selected category filter. Enables the breed filter selection |
| Breed filter is clicked | Options of breed that match both the pre-selected category and size filters are displayed in a dropdown menu |  |  | Displays the animal breed options that match both the pre-selected category and size filters in a dropdown menu format |
| Breed option item is clicked | List of animals displayed on the screen changes to match all filters applied (category, size, and breed) |  |  | Displays the list of animals that match all filters selected (category, size, and breed) |
| Animal list item is clicked | Animal profile screen opens |  |  | Animal profile screen is displayed |
| Send intent form button is clicked | Prompts the user a selection of e-mail providers available on the user’s phone that may be used to send the intent form |  |  | A selection of e-mail providers available on the user’s phone is prompt so the user can choose which one to use to send the intent form |
| E-mail provider is selected | Opens the external e-mail application with automatically filled in recipient address, subject and e-mail body with animal information |  |  | External e-mail application is opened with automatically filled in recipient address, subject and e-mail body with animal information |
| Shelter name TextView is clicked | Shelter profile screen opens |  |  | Shelter profile screen is displayed |
| Zoom in button on the integrated Google Map is clicked | Map is zoomed in one level, being able to zoom in up to the level 20 |  |  | Zoom in one level is applied on map, being able to zoom in up to the level 20 |
| Zoom out button on the integrated Google Map is clicked | Map is zoomed out one level, being able to zoom out up to the level 7 (starts on the minimum zoom level) |  |  | Zoom out one level is applied on map, being able to zoom out up to the level 7 (starts on the minimum zoom level |
| Go back to animal profile button is clicked | Returns to animal profile screen the user navigated from |  |  | Go back to the animal profile screen the user navigated from |
| Back button is clicked | Return to the previous screen |  |  | Go back to the previous screen |

**5.7 Chapter Summary**

Three major Agile Testing methodologies were presented in this chapter (White-Box testing, Black-Box Testing and Grey-Box Testing) as well as their advantages and disadvantages in order to decide on which one to be used within the current project. After deep analyse of each option, it was proven that the Grey-Box testing is the most appropriate since it combines both Black and White-Boxes advantages and favours our time frame.

It is also presented in this chapter the Successful Testing process for this application completed as well as all final tests that were passed.

The next chapter will present the end-user testing of the software as well as their feedback and a conclusion for this project.

**References**

GeeksforGeeks, 2020. *GeeksforGeeks.* [Online]   
Available at: https://www.geeksforgeeks.org/gray-box-testing-software-testing/  
[Accessed 18 June 2020].

Jain, M., 2020. *GeeksforGeeks.* [Online]   
Available at: https://www.geeksforgeeks.org/differences-between-black-box-testing-vs-white-box-testing/  
[Accessed 18 June 2020].

JavaTPoint, 2020. *JavaTPoint.* [Online]   
Available at: https://www.javatpoint.com/grey-box-testing  
[Accessed 18 June 2020].

Niesse, E., 2018. *HRchitect.* [Online]   
Available at: https://hrchitect.com/testing-phase-clients-role-implementation/  
[Accessed 16 June 2020].

reqtest, 2018. *Reqtest.* [Online]   
Available at: https://reqtest.com/testing-blog/agile-testing-principles-methods-advantages/  
[Accessed 17 June 2020].

Software Testing Help, 2020. *Software Testing Help.* [Online]   
Available at: https://www.softwaretestinghelp.com/black-box-testing/  
[Accessed 18 June 2020].

STFa, 2020. *Software Testing Fundamentals.* [Online]   
Available at: http://softwaretestingfundamentals.com/white-box-testing/#:~:text=WHITE%20BOX%20TESTING%20(also%20known,is%20known%20to%20the%20tester.  
[Accessed 17 June 2020].

STFb, 2020. *Software Testing Fundamentals.* [Online]   
Available at: http://softwaretestingfundamentals.com/black-box-testing/#:~:text=BLACK%20BOX%20TESTING%2C%20also%20known,%2Dfunctional%2C%20though%20usually%20functional.  
[Accessed 18 June 2020].

STFc, 2020. *Software Testing Fundamentals.* [Online]   
Available at: http://softwaretestingfundamentals.com/gray-box-testing/  
[Accessed 18 June 2020].