**COMP3505 - Software Testing**

**Assignment. Report #1 – Introduction to Testing and Defect Tracking**

|  |  |
| --- | --- |
| Group #: | 4 |
| Student Names: | Kevin Ruiz |
| Charlemagne Palisoc |
| Ryan Wong |

# **Answer the following questions, each in one paragraph (20%)**

1. **Based on your experience, please describe the advantages and disadvantages of exploratory testing?**

Exploratory testing is an approach to software testing that is not rigid and encourages the creativity of the tester. It is advantageous because it utilizes the tester’s existing knowledge and personal skills making them more involved as they conduct the test. It is also a great way to learn what the application can do, as the tester conducts the test, they are learning in parallel, they gain knowledge and expertise with using the application they can bring more value and provide better feedback on improving the system. ET also requires less preparation, there is not a lot of documentation needed to begin the test. ET is also great at providing rapid feedback because the tester can focus and spend time exploring and finding bugs within the system and communicate them to the developers.

Exploratory testing is good that it does not constrain you to follow particular testing, but gives a tester some freedom on how they approach testing. However, this could also be a disadvantage because the tester can spend too much time testing one area at the exclusion of others if they do not establish a goal/high-level plan for the test. It is also sometimes not easy to say which tests were already performed. It is also difficult to document each procedure of the tests, because the tester will be creating the tests as they go, and depending on the results of the previous test will decide on how to proceed.

1. **Based on your experience, please describe the advantages and disadvantages of scripted testing?**

Scripted testing is an approach where the testers follow a *script* that includes well-documented test cases and test steps. Its advantages are the unchallenging ways of finding functional defects as well the having that standardized documentation, which helps in repeatability and tracking.

Finding fewer bugs is one of its huge disadvantages, this is because it is focused on limited areas constrained by the script and its “fixed” expected output.

1. **Which one of these tests, scripted vs exploratory should be automated?**

Exploratory testing is a type of software testing that is designed and executed at the same time. Hence, it is not recommended for automated testing, it encourages the tester to be more creative and active when doing the testing. Scripted testing, on the other hand is when tests are designed and recorded, which can then be executed at a later time. Scripted tests can be automated easily using the recorded test cases which can include what part of the software to be tested, and what inputs and outputs are to be used in testing the functionalities of the application.

1. **What is the objective of regression testing and when does this type of test usually happen?**

The objective of regression testing is to ensure that the changes made to the software to correct an identified bug do not cause new bugs in previously tested cases of the validated software. In essence, regression testing looks for “any unexpected side-effects” of bug corrections, and asks - are the previous functions of the system still working after the changes we made? In theory, regression testing should happen every time a program is modified.

1. **What is the difference between bug tracking systems and issue tracking systems?**

A bug is an error in the coding of the software, a failure that creates an incorrect or unexpected output and/or causes the system to behave in an unexpected way. A bug tracking system is a software system or some other method that keeps track and reports software bugs in a developing software project.

An issue is a problem with functionality and can include a lack of documentation, a missing feature, or a bug. An issue tracking system is software or some other method that keeps track and reports functionality problems in a software project.

1. **What are the downsides of writing multiple test cases in one case, such as Test case 14, and 15?**

By writing multiple test cases in one case information and documentation about the testing of those cases can be missed, lost or misunderstood. It is better to practice to write a test case for every test to provide specific details for the state of the software, the testing input used, and the outputs. When doing several tests in one case, there is a risk of altering the state of the system with each iteration of the test and creating false assumptions about what state the system is in. There is no way to guarantee that entering input in the field will not affect future outputs for the multiple test case inputs that will be run. This kind of multiple test cases introduces new variables to the testing, that may be hard to go back and recreate or determine.

# **Give a description of your High-level plan for Your Exploratory Testing (10%)**

* The application to be tested is an online PetStore, hence we think that testing the applications ordering and purchasing features is very important. Testing this functionalities will touch on several functionalities of the application as well such as:
  + account creation
  + sign-in/sign-out
  + store catalog (browsing/searching of pets)
  + cart (add to cart, modify/update cart)
  + payment and billing information
  + order/checkout summary
* With a goal in mind, which is to order a pet, we are able to touch on different functionalities of the application, diggin into each functionality.
* From there we can explore the features such as changing account information (i.e changing name/password, etc.), modify and update carts (i.e testing max/min quantity of pets that can be bought), browsing pet catalogues and using the search functionality to find specific pets, modifying address information, and confirming order details and summary.
* While going through the steps of placing an order we intend to see if the functionalities that leads to ordering a pet are working as they should.

# **Design test cases for the order/purchase functionality, and include them here: (12%)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case # | Use Case | Function Being Tested | Initial System State | Input | Expected Output |
| 1 | Order | Checkout | Application is open.  User is logged in.  One item in cart.  Proceed to checkout. | Fill in the payment details, and billing address correctly.  Continue. | User receives order confirmation and summary of the order. |
| 2 | Order | Checkout | Application is open.  User is logged in.  One item in cart.  Proceed to checkout. | In Billing Address section. Leave one of the required fields empty (first name, last name, address1, city, state, zip, country)  Continue. | Error message is shown to user, required fields cannot be empty. |
| 3 | Order | Checkout | Application is open.  User is logged in.  Add one item in cart.  Proceed to checkout. | In Payment Details section. Leave Card number empty.  Continue. | Error message is shown to user, required field cannot be empty. |
| 4 | Order | Checkout | Application is open.  User is logged in.  Add one item in cart.  Proceed to checkout.  Fill in payment details and billing address correctly.  Check Ship to different address box.  Continue. | In Shipping Address section. Change the name and address to different values, making sure that the required fields are correctly inputted.  Continue.  Confirm. | User is shown confirm window for information. Clicking continue successfully submits the order. |
| 5 | Order | Checkout | Application is open.  User is logged in.  One item in cart.  Proceed to checkout.  Fill in payment details and billing address correctly.  Check Ship to different address box.  Continue. | In Shipping Address section. Leave one of the required fields empty (first name, last name, address 1, city, state, zip, country)  Continue. | Error message is shown to user. Required fields cannot be empty. |
| 6 | Order | Checkout | Open application.  Make sure you are not logged in.  Add an item to cart. | Proceed to checkout. | User will be redirected to login first. Error message shown. |
| 7 | Order | Checkout | Open application.  Make sure you are not logged in.  Add one or two item to cart.  Proceed to checkout.  User is prompted to login. | Fill in username and password. Sign in | User should be taken to payment and billing details for order checkout. |

# **One of the important tasks in the testing process is to improve and update your test cases. What improvements would you make to test CASES 1-41? (5%)**

* Test cases that include multiple input field testing should each be created as separate test cases.
* Test cases steps could be numbered to ensure order of completion.
* Some test cases have, in our opinion, have invalid expected outcomes. For example, test case #33’s changing item quantity in cart to 0 should have an expected outcome where the item must be removed from the cart, instead of showing an error message.
* Test case #37 and #38 are the same test cases, with just a little bit of change in #38. Similar test cases like this since they are consecutive, can instead refer to the previous case, then let the tester know to repeat those steps, then do the following new steps.

# **Difficulties encountered, challenges overcome, and lessons learned (3%)**

Organization is obviously a key tool when completing software testing. Standardizing our bug reporting information and format to ensure all team members were recording in the same manner with the same language was a huge lesson. Establishing a standard bug reporting, and communicating this to the tester within the team is important, this way bugs are consistently created. Some difficulty in recreating bugs discovered by other testers was encountered, but communication and collaboration overcame this. When performing exploratory testing, there is a likelihood that a tester will have focused on one particular functionality of the system, essentially ignoring the rest of the application. We found that having a high level plan on approaching the exploratory testing, can mitigate and help the tester overcome this.

# **Comments/feedback on the assignment document itself (+2%)**

Assignment document was very helpful in describing what the assignment is all about. It contained all the necessary information to get the students started with the assignment, and how they should approach the assignment. The assignment rubric is a nice addition providing the students with an idea of how they will be graded, as well as the expected files to be submitted. The document may also need some improvement, with the amount of information included we found it intimidating the first time. Moving some parts of the document into its own separate file, like the scripted testing and rubric. Overall, the document itself was very helpful during the assignment, and can serve to be a great reference for studying in the future.