GUI TESTING

OOAD | UNIT 5

Introduction

- Graphical User Interface (GUI)of an application is what the user sees.
- GUI testing is done to remove bugs in the user interface, to make it more user-friendly and to keep the designs minimalistic unless required otherwise.
- GUI testing is a process to test application's user interface and to detect if application is functionally correct. GUI testing involves carrying set of tasks and comparing the result of same with the expected output and ability to repeat same set of tasks multiple times with different data input and same level of accuracy.

What does GUI testing include?

- GUI Testing includes :
- how the application handles keyboard and mouse events,
- how different GUI components like menu bars, toolbars, dialogs, buttons, edit fields, list controls, images etc. react to user input and whether or not perform in the desired manner.
- Implementing GUI testing for your application early in the software development cycle speeds up development, improves quality and reduces risks towards the end of the cycle. GUI Testing can be performed both manually with a human tester or could be performed automatically with use of a software program.

The following checklist will ensure detailed GUI Testing.

- Check all the GUI elements for size, position, width, length and acceptance of characters or numbers. For instance, you must be able to provide inputs to the input fields.
- Check you can execute the intended functionality of the application using the GUI
- Check Error Messages are displayed correctly
- Check for Clear demarcation of different sections on screen
- Check Font used in application is readable
- Check the alignment of the text is proper
- Check the Color of the font and warning messages is aesthetically pleasing
- Check that the images have good clarity
- Check that the images are properly aligned
- Check the positioning of GUI elements for different screen resolution.

GUI testing can be done through three ways:

Manual/Human tester

Record and Replay

Model based

Model based GUI testing

- **Event-based model -** Based on all events of the GUI need to be executed at least once.
- **State-based model -** "all states" of the GUI are to be exercised at least once.
- **Domain model -** Based on the application domain and its functionality.
- **GUI testing tools** serve the purpose of automating the testing process of software with graphical user interfaces. Example: **Oracle Application Testing Suite** is a comprehensive, integrated testing solution for Web applications, Web Services, packaged Oracle Applications and Oracle Databases.

When should tests be automated?

- Functionality should always be understood and validated manually first
- A test should not be too difficult (expensive) to automate
- The team should be staffed with skilled automation developers
- High priority customer scenarios are identified
- The product code is stable and most tests won't need constant updating
- The test team is staffed to analyse test results

Challenges in GUI Testing

- The most common problem comes while doing regression testing is that the application GUI changes frequently. It is very difficult to test and identify whether it is an issue or enhancement. The problem manifests when you don't have any documents regarding GUI changes.
- Test case generation: What combinations of user actions to try?
- Oracles: What is the expected GUI behaviour?
- Coverage: How much testing is enough?
- Regression testing: Can test cases from an earlier version be re-used?
- Representation: How to represent the GUI to handle all the above?

Two approaches to GUI testing

- **1.Black box**
- Launch application
- Simulate mouse and keyboard event
- Compare final look to an existing screen dump
- Very brittle test cases
- Cannot test business model
- Framework independent
- **2. Glass box**
- Launch application in the testing code
- Obtain references to the various components and send events to them
- Assert the state of components directly
- Test cases harder to break
- Business model can be tested
- Framework dependent

- RESOURCES:
- http://www.guru99.com/gui-testing.html
- http://www.tutorialspoint.com/software_testing_dictionary/use_interface_testing.htm
- http://softwaretestingexpertview.blogspot.in/2012/06/explain-gui-testing-and-its-contents.html
- http://www.softwaretestinghelp.com/sikuli-tutorial-part-1/