

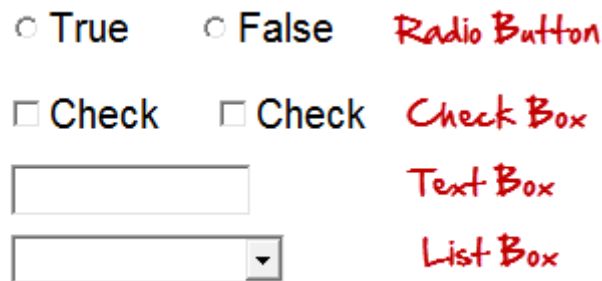
Complete Guide for GUI Testing

There are two types of interfaces in a computer application.

Command Line Interface is where you type text and computer responds to that command.

GUI stands for Graphical User Interface where you interact with the computer using images rather than text.

Following are the GUI elements which can be used for interaction between the user and application:



GUI Testing, is validation of above elements.

In this tutorial, you will learn-

- [What is GUI Testing?](#)
- [Need of GUI Testing](#)
- [What do you Check in GUI Testing?](#)
- [Approach of GUI Testing](#)
- [GUI Testing Test Cases](#)
- [Demo: How to conduct GUI Test](#)
- [Challenges in GUI Testing](#)

What is GUI Testing?

Graphical User Interface (GUI) testing is the process of testing the system's GUI of the System Under Test. GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars - toolbar, menu bar, dialog boxes and windows, etc.

GUI is what user sees. Say if you visit guru99.com what you will see say home page it is the GUI (graphical user interface) of the site. A user does not see the source code. The interface is visible to the user. Especially the focus is on the design structure, images that they are working properly or not.

The screenshot shows the Guru99 website home page. At the top, there is a search bar with the text "Search" and a "Google" logo. Below the search bar, there is a large orange arrow pointing to the right, towards two icons: a Java logo (a red flame) and a Linux logo (a penguin). The main content area features the text "experience." followed by "Here you learn by practice. We make tons of efforts to take boredom out of learning and make education a fun experience." and "Inside, you will find tons of video tutorials". Below this is a horizontal line and the text "All provided FREE!!!". The "Tutorials Library" section is divided into three columns: "TESTING", "SAP", and "LIVE PROJECTS". Each column contains a list of tutorial links with a play button icon.

TESTING	SAP	LIVE PROJECTS
Learn Software Testing	Learn SAP Beginner	Live Testing Project
QTP (Quick Test Professional)	Learn SAP ABAP	Live Selenium Project
Learn Selenium	Learn SAP HR/HCM	Live Ecommerce Project
Learn Mobile App Testing	Learn SAP FICO	Live UFT Testing

In above example, if we have to do GUI testing we first check that the images should be completely visible in different browsers.

Also, the links are available, and the button should work when clicked.

Also, if the user resizes the screen, neither images nor content should shrink or crop or overlap.

Need of GUI Testing

Now the basic concept of GUI testing is clear. The few questions that will strike in your mind will be

- Why do GUI testing?
- Is it really needed?
- Does testing of functionally and logic of Application is not more than enough??Then why to waste time on UI testing.

To get the answer think as a user, not as a tester. A user doesn't have any knowledge about XYZ software/Application. It is the UI of the Application which decides that a user is going to use the Application further or not.

A normal User first observes the design and looks of the Application/Software and how easy it is for him to understand the UI. If a user is not comfortable with the Interface or find Application complex to understand he would never going to use that Application Again. That's why, GUI is a matter for concern, and proper testing should be carried out in order to make sure that GUI is free of Bugs.

What do you Check in GUI Testing?

The following checklist will ensure detailed GUI Testing.

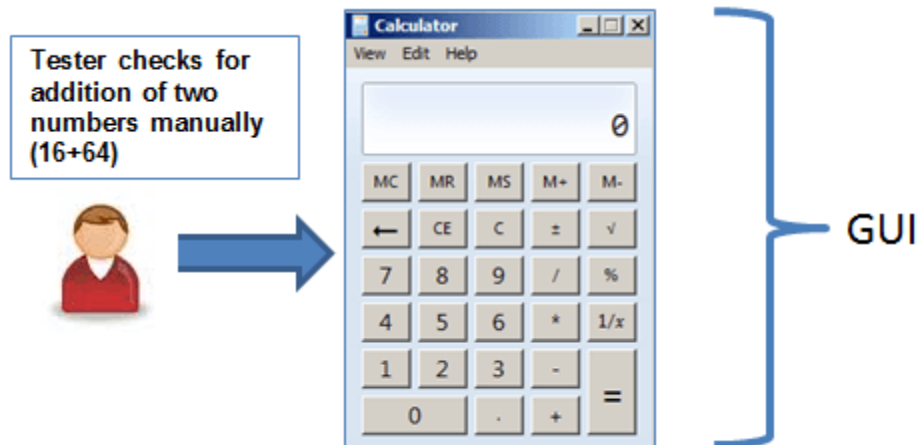
- Check all the GUI elements for size, position, width, length and acceptance of characters or numbers. For instance, you must be able to provide inputs to the input fields.
- Check you can execute the intended functionality of the application using the GUI
- Check Error Messages are displayed correctly
- 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.

Approach of GUI Testing

GUI testing can be done through three ways:

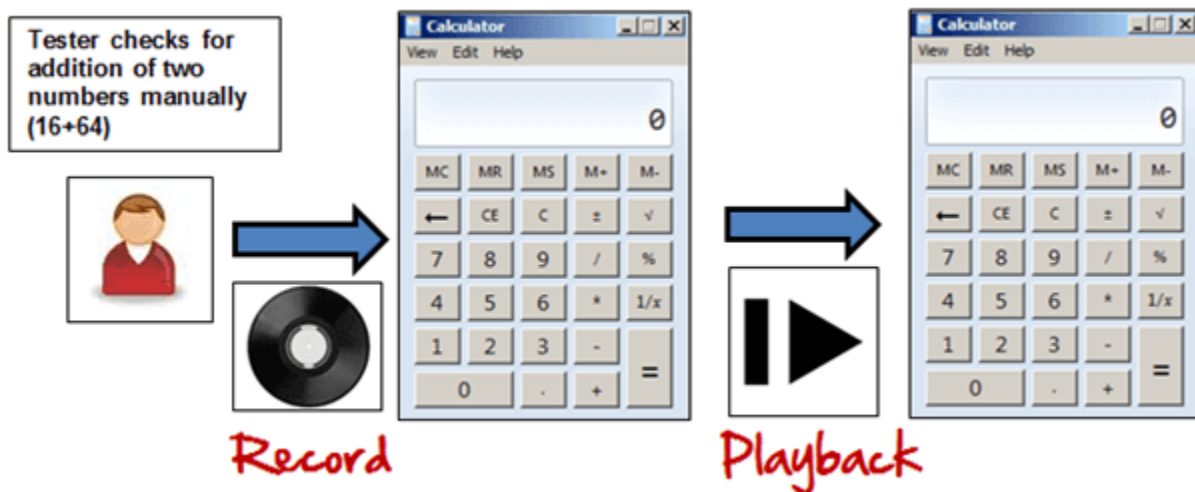
Manual Based Testing

Under this approach, graphical screens are checked manually by testers in conformance with the requirements stated in the business requirements document.

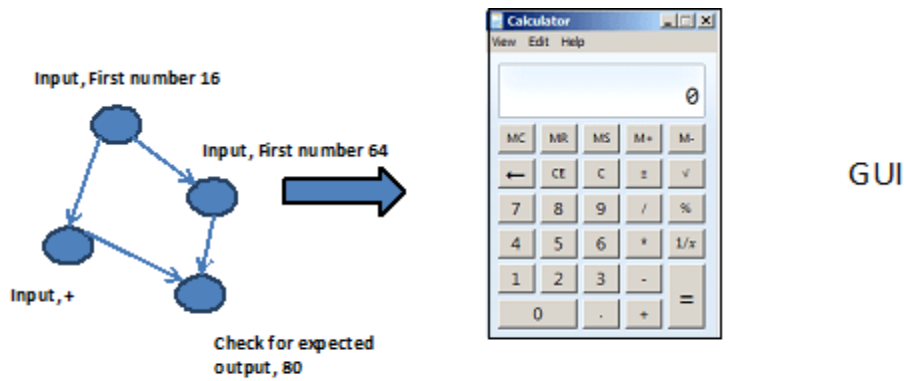


Record and Replay

GUI testing can be done using automation tools. This is done in 2 parts. During Record, test steps are captured by the automation tool. During playback, the recorded test steps are executed on the Application Under Test. Example of such tools - [QTP](#).



Model Based Testing



Model Based Testing

A model is a graphical description of system's behavior. It helps us to understand and predict the system behavior. Models help in a generation of efficient test cases using the system requirements. Following needs to be considered for this model based testing:

- Build the model
- Determine Inputs for the model
- Calculate expected output for the model
- Run the tests
- Compare the actual output with the expected output
- Decision on further action on the model

Some of the modeling techniques from which test cases can be derived:

- Charts - Depicts the state of a system and checks the state after some input.
- Decision Tables - Tables used to determine results for each input applied

Model based testing is an evolving technique for the generating the test cases from the requirements. **Its main advantage, compared to above two methods, is that it can determine undesirable states that your GUI can attain.**

Following are open source tools available to conduct GUI Testing.

Product	Licensed Under
AutoHotkey	GPL
Selenium	Apache

Sikuli	MIT
Robot Framework	Apache
Water	BSD
Dojo Toolkit	BSD

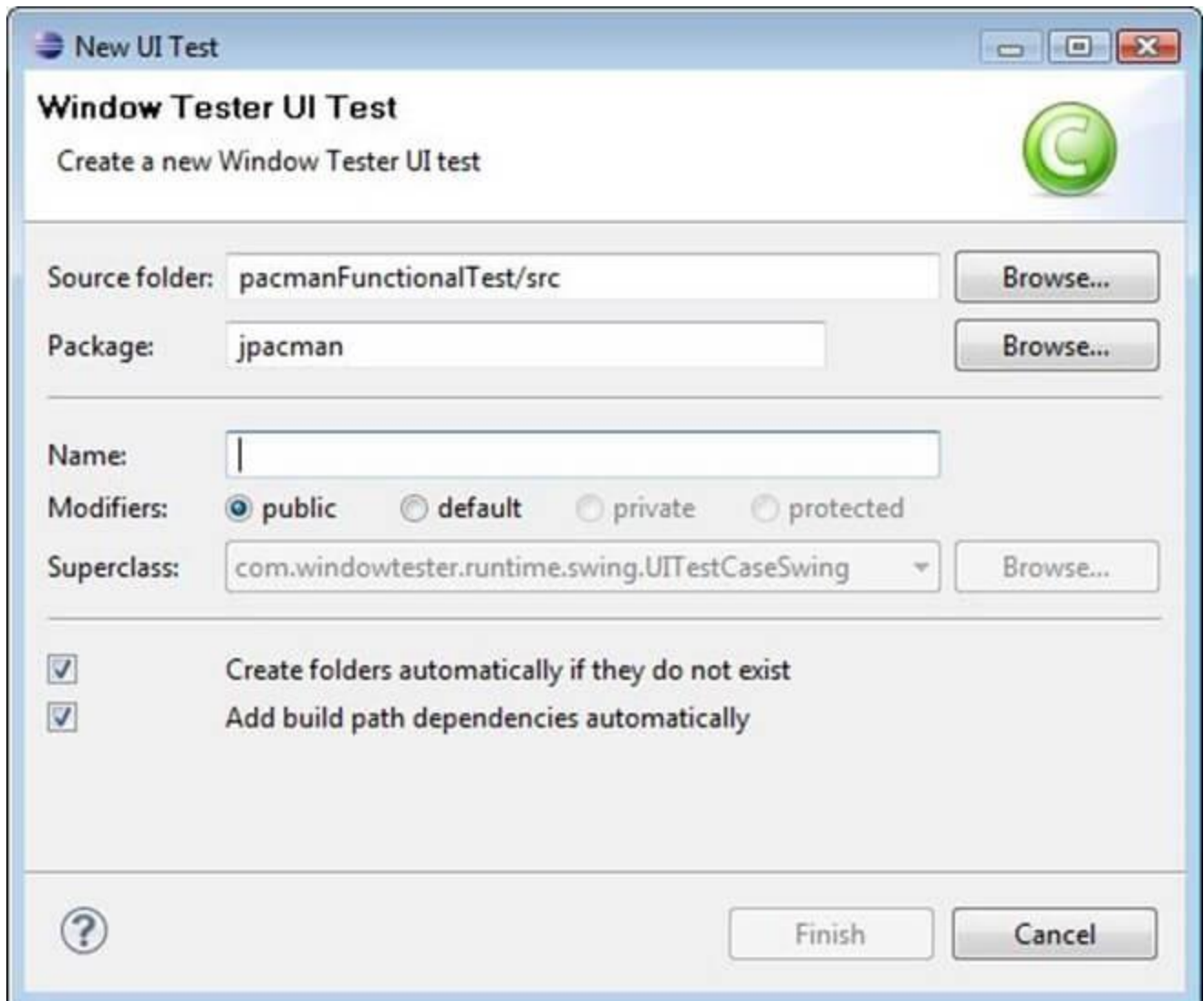
GUI Testing Test Cases

GUI Testing basically involves

1. Testing the size, position, width, height of the elements.
2. Testing of the error messages that are getting displayed.
3. Testing the different sections of the screen.
4. Testing of the font whether it is readable or not.
5. Testing of the screen in different resolutions with the help of zooming in and zooming out like 640 x 480, 600x800, etc.
6. Testing the alignment of the texts and other elements like icons, buttons, etc. are in proper place or not.
7. Testing the colors of the fonts.
8. Testing the colors of the error messages, warning messages.
9. Testing whether the image has good clarity or not.
10. Testing the alignment of the images.
11. Testing of the spelling.
12. The user must not get frustrated while using the system interface.
13. Testing whether the interface is attractive or not.
14. Testing of the scrollbars according to the size of the page if any.
15. Testing of the disabled fields if any.
16. Testing of the size of the images.
17. Testing of the headings whether it is properly aligned or not.
18. Testing of the color of the hyperlink.

Demo: How to conduct GUI Test

Here we will use some sample test cases for the following screen.



Following below are the example of the Test cases, which consists of UI and Usability test scenarios.

TC 01- Verify that the text box with the label "**Source Folder**" is aligned properly.

TC 02 - Verify that the text box with the label "**Package**" is aligned properly.

TC 03 – Verify that label with the name "**Browse**" is a button which is located at the end of TextBox with the name "**Source Folder**."

TC 04 – Verify that label with the name "**Browse**" is a button which is located at the end of Text Box with the name "**Package**."

TC 05 – Verify that the text box with the label "**Name**" is aligned properly.

TC 06 – Verify that the label "**Modifiers**" consists of 4 radio buttons with the name public, default, private, protected.

TC 07 – Verify that the label "**Modifiers**" consists of 4 radio buttons which are aligned properly in a row.

TC 08 – Verify that the label "**Superclass**" under the label "**Modifiers**" consists of a dropdown which must be proper aligned.

TC 09 – Verify that the label "**Superclass**" consists of a button with the label "**Browse**" on it which must be properly aligned.

TC 10 – Verify that clicking on any radio button the default mouse pointer must be changed to hand mouse pointer.

TC 11 – Verify that user must not be able to type in the dropdown of "**Superclass.**"

TC 12 – Verify that there must be a proper error generated if something has been mistakenly chosen.

TC 13 - Verify that the error must be generated in the RED color wherever it is necessary.

TC 14 – Verify that proper labels must be used in the error messages.

TC 15 – Verify that the single radio buttons must be selected by default every time.

TC 16 – Verify that the TAB button must be work properly while jumping on other field next to previous.

TC 17 – Verify that all the pages must contain the proper title.

TC 18 – Verify that the page text must be proper aligned.

TC 19 – Verify that after updating any field a proper confirmation message must be displayed.

TC 20 - Verify that only 1 radio button must be selected and more than single checkboxes may be selected.

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.

Conclusion:

Success of a software product immensely depends on how the GUI interacts with the user and ease in using its various features. Hence, GUI testing is very important. Manual GUI testing can sometimes be repetitive and boring and hence error prone. Automation is highly recommended for GUI testing.