Special Types of Testing

Introduction

- Testing a system is meant to ensure an error-free, quality product.
- A system undergoes several tests before its launch.
- In this module we look at configuration testing and graphical user interface testing.

Introduction

- Testing a system is a methodology, which involves both the hardware and the software components of the system.
- System testing falls under the scope of black box testing.
- System testing does not require a software tester to have a thorough knowledge about the internal design and code of the software. It unravels the defects within the system and between the various links in the code (assemblages).

Introduction

- In system testing,
- The system that is to be tested is configured in a controlled environment.
- Real life scenarios are simulated in the test environment.
- This type of testing is considered to be complete on two scenarios: First when the actual results and the expected results match, and Second when the differences between the actual results and the expected results are well explained and accepted based on client input.

Introduction

- System testing is considered to be a process of exploring the functionality of the system and identifying the faults within the system. The following example gives you a better idea about **breaking the system.**

System testing

- Example
- Testers input **Name of Country** in a text box that has been designed to accept only **Name of City**.
- This is done to check the system's response for incorrect data with reference to **breaking the system**.

Configuration Testing

- Configuration testing involves testing the various configuration possibilities for a computer used at home or in an organization.
- first testing task assigned to a software tester. As a tester you need to ensure that the software works fine without any problem, for all possible hardware combinations.
- **Testing the system** would be a very simple task, when the hardware combinations of computers are identical. In addition, there would be no confusion with the option buttons available to click, and components would interface perfectly every time you use the system.

Configuration Testing

- The cost of performing configuration testing is very low, but the benefits are large due to the repeated tests.
- Configuration testing is always considered as a cost effective method.

Overview of Configuration Testing

- configuration testing would be checking the software's functioning ability with various hardware configurations.
- Configuration testers test printers, Network Interface Cards (NICs), and so on. Configuration testing is also known as portable testing or hardware compatibility testing.

Configuration Testing

- Hardware configuration elements that you may have to test
- ***Manufacturers:*** Computer manufacturers either design their own computers or obtain certain components from a third party manufacturer to build a computer. Some people also assemble their computers using off-the-shelf components available in the market.
- ***Components:*** Computers are made up of various components such as system boards, component cards, network cards, disk drives, CD-ROM and DVD drives, video cards, sound cards, input/output cards, and much more specialized hardware for advanced use.
- ***Peripherals:*** Peripherals are the external hardware devices such as printers, scanners, mouse devices, keyboards, monitors, fax modems, cameras, and joysticks that are plugged into the computer.
- ***Interfaces:*** Interfaces are the components and peripherals that are plugged into a computer through various internal and external connectors.

Configuration testing

- ***Options and Memory:*** Components and peripherals with various available hardware options and memory sizes can be bought today. You also have the privilege of upgrading printers to support extra memory and speed up the printing process.
- ***Device Drivers:*** Device drivers are the drivers that help in establishing communication between hardware components and software applications. The device drivers are provided by the hardware manufacturers and are installed in the computer.
- **Isolating Configuration Bugs**
- A software tester must be aware of the common bugs that can arise in the process of testing. Configuration bugs are found by performing similar kind of operations on different computers with different hardware setups.
- Discovery of configuration bugs can cost you a lot and hence you must make efforts to detect them during the early stages of testing.

Configuration Testing

- **Sizing up the Job**
- The job of a configuration tester is very challenging. The number of software testers required for testing each of the tasks is first addressed. Consider testing a gaming software application on Microsoft Windows operating system. In this scenario, it is necessary to check for appropriate sound effects. So, configuration testing is carried out with the various sound cards and graphics cards, along with the modem specifications.

Identifying Software Configuration

- Configuration testing ensures that the software works perfectly fine with the various hardware configurations.
- Only the most important features that are different from one another are tested using equivalence partitioning.

Example

- **Equivalence Partition**
- Equivalence partitioning, also called as equivalence classing, is a process of classifying the test cases and grouping them into different classes. This method helps to reduce test cases to a finite number of test cases without compromising on the quality of the test being carried out.
- Consider a text box that accepts numeric values ranging from 18 and 60 (18 and 60 are part of the equivalence class).

Valid Classes:

Values from 18 to 60 Invalid Classes:

Values < 18

Values >60

Values such as 19, 24, and 59 fall under the valid class, while values like 17 and 75 fall under the invalid class.

Design the Test Cases to Run on Each Configuration

- A tester must have a good understanding of the product. Black box testing provides better knowledge about the product to the tester. It thereby enables the tester to learn about the features of the product. Hence, test cases are written before configuration testing. Let us now learn some of the steps required to test each configuration.

1. Select the object to be tested.
2. Set up the test configuration from the list.
3. Start the software.
4. Load in the file configtest.doc.
5. Ensure that the displayed file is correct.
6. Print the document.
7. Confirm that there are no error messages and that the printed document matches the standard document.
8. Log any discrepancy as a bug.

Deciding the Hardware Configuration

- Having an idea about the products and their manufacturers enables the testers to decide the hardware configuration. Based on the hardware configuration, the equivalence partitions are designed and the standards to be followed are identified
- *The Apple website provides you with information on how to develop and test hardware devices for Apple computers. The Apple website also includes the links pertaining to test-labs with information on conducting configuration testing*.

Configuration Testing other Hardware

- Configuration testing involves checking the configurations of the various software and hardware that support a system. Apart from knowing the type of hardware, you also need to know the memory size, CPU speed, and so on and ensure that these hardware types connect to the correct piece of hardware and software configuration.
- **Some of the common questions that one needs to answer before performing configuration testing are as follows:**

1. What external hardware will operate with the software?
2. What models and versions of the hardware are available?
3. What features or options does the hardware support?

Configuration Testing

- While performing configuration testing, one must ensure to have relevant set of questions to learn about the system to be tested.
- Equivalence class partitions of the hardware are created based on input obtained from the people who work with the project manager. Based on their input, test cases are developed and they are run on the appropriate hardware to be tested.

Graphical User Interface Testing

- The Graphical User Interface (GUI) is the front-end that acts as an interface for the users of computer or an electronic gadget.
- A software product is developed and designed for use by its customers. Be it office, home or school, you can find several software products used on a regular basis. Therefore, it becomes necessary to check the usability of the software products.
- Usability of a software product can be seen in the appropriate functional and effective interaction of the software product. It can be said that usability is very similar to Ergonomics.
- *Ergonomics relates to the science of designing things in such a way that makes the product easy to use it.*

Graphical User Interface Testing

- a software tester, would be the first person to use the software product. Usability of one product differs from that of another. The medium through which you interact with the software is known as User Interface or UI. There are some peculiar cases, one of which is discussed in the following example

Graphical User Interface Testing

- Some of the reasons for the popularity of the GUI are:

1. It is easy to understand visual interface.
2. It is flexible to use in most of the application areas.
3. It is helpful for people who have difficulty in typing.
4. It provides visibility of multiple windows, to handle information in a better way.
5. It helps in controlling the screens as per user's choice.
6. It facilitates exchange of information because of integration of the packaged and customized applications.

Graphical User Interface Testing

- Although GUI has simplified things for users, it has complicated them for a developer.
- GUI testing can be performed either manually or automatically.
- Manual testing is a time consuming process, which is performed without the help of automated tools like winruuner, silk test, and Quick Test Professional (QTP).
- Automated testing is performed by using automated tools like load runner, winruuner and Quick Test Professional (QTP).

Graphical User Interface Testing

- GUI testing is commonly known as usability testing or user interface testing.
- Usability testing is the process of checking a product's compatibility, when in use. After completion of the testing process, the software product is released to a set of users as a beta version or a pre-release version. The users evaluate and assess the performance of the software based on customer experience

GUI Testing Stages

Stages	Testing Involved
Low Level Stage	Checklist testing Navigation
Application	Equivalence partitioning Boundary values Decision tables State transition testing
Integration	Desktop integration Communication Synchronization
Non-Functional	Soak testing Compatibility testing Platform/environment

General guidelines for GUI testing

- All the dialog boxes have a consistent appearance throughout the application system.
- Every field on the screen has an associated label.
- Every screen has an equivalent **OK** and **Cancel** button (with an appealing color combination).
- Every field in the dialog box supports short cut key functioning.
- Tab order is set horizontal to the fields, as sometimes it can be vertical.
- Mandatory fields are marked with red asterisk (*) to denote that they are mandatory.
- Default key <Enter> is set as **OK** for the dialog box.
- Default key <Esc> is set as **Cancel** for the dialog box.

Ten points while performing GUI testing:

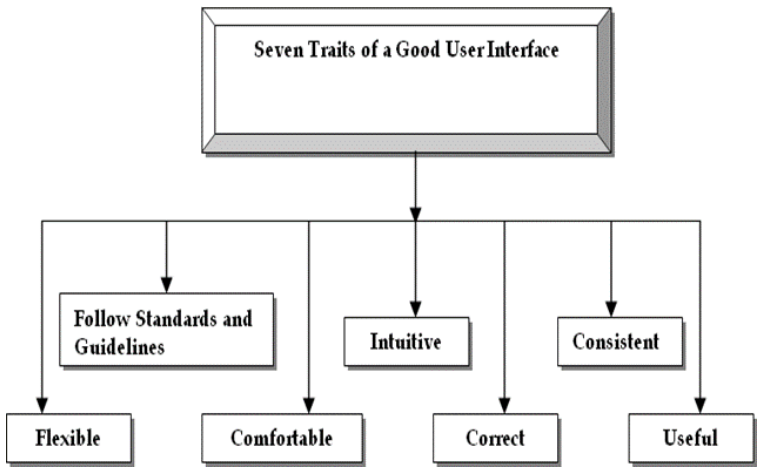
1. Test for user friendly labels, messages, related message content, and understandability of the message.
2. Test ease of navigation.
3. Test availability of help for a particular operation.
4. Test use of colors, fonts, alignment, and tab orders.
5. Test end to end navigation.

Ten points while performing GUI testing:

6. Test functionality of control objects like buttons, textbox, list box, and so on.
7. Test for the risk of critical defects.
8. Test the strict adherence to time and schedule.
9. Test the risk of Quality Assurance (QA) resources yet to be familiarized on the new application.
10. Test the risk of low priority aspects of the application that is not being tested till the later stages.

Standards and Guidelines

- A software tester needs to pretend to be the user while testing for usability and locate the possible errors or problem-prone areas in the software product.



Important traits of a good user interface

- ***Follow Standards and Guidelines:*** It is very important for every software product to adhere to standards and guidelines. When the software is running on Windows platform, it is assumed that it is adhering to a set of standards.

Important traits of a good user interface

- the first personal computer released in 1975 user interfaces were switches and lights.
- This computer was created for hobbyists. I
- In today's world, a customer looks for more in every software product. In view of the customer demands, we need to ensure that the following points are considered while performing user interface testing.

Important traits of a good user interface

- Check if the user interface is clean, unobtrusive, and not cluttered with options and information. The user interface must not get in the way of what you want to perform or the functions you need, and the expected response must be obvious.

Important traits of a good user interface

- Check if the user interface is organized and laid out well. Ensure that it allows you to navigate from one function to another. At any given time, you should be able to do nothing, back up, or back out.

- Check for excessive functionality. Ensure that the software does not try to do too much, either as whole or as a part.
- Check if the **help system** really helps when everything fails.

Important traits of a good user interface

- ***Consistent:*** Consistency within the software and with other software is a key att**ribute**.
- Inconsistencies when moving from one program to another frustrate the users. Thus, it is necessary to follow a standard for the software or the platform, else attention must be paid to the features of the software to ensure that similar operations are performed in the same way.
- The consistency criteria to be outlined while performing usability testing are:
- ***Shortcut Keys and Menu Selections:*** The shortcut keys are similar to accessing **Help** by pressing **F1** in Windows.
- ***Terminology and Naming:*** You need to look for same terms used throughout different versions of the software and check whether the features are named consistently.
- ***Audience:*** You need to check whether the software consistently addresses all kinds of audience level.
- ***OK and Cancel Button Locations:*** You need to check whether the location of the buttons remains same from one platform to another.

Important traits of a good user interface

- ***Flexible:*** Flexibility relates to the ease with which the user performs tasks as per requirements.
- Eg A calculator having both scientific and standard normal view.
- Flexibility in software provides the following features:

a) State Jumping: Software which is very flexible, gives more options to accomplish the same task.

b) State Termination and Skipping: Software with power-user modes allows the user to skip numerous prompts or windows and go to the destination directly.

c) ***Data Input and Output:*** Users demand ways in which they can enter data and view their results.

Important traits of a good user interface

- ***Comfortable:*** The user must be comfortable using the software.

- ***Appropriateness:*** The software designed must have a proper look, feel, and relate to what it is supposed to do.
- ***Error Handling:*** Programs must be written in such a way that they warn users before a critical operation and also allow the users to restore the lost data.
  - Eg The **Undo** feature
- ***Performance:*** Performance does not refer to speed, instead it implies that more than one program can flash error messages at a greater speed.
  - Eg The **Status** bars that display the accomplishment of a task.

Important traits of a good user interface

- ***Correct:*** Testing for correctness implies checking whether the user interface accomplishes what it is supposed to do. The appearance of the **hourglass** symbol indicates that the software is busy and cannot accept any input at that particular time.

(7) While testing, a tester needs to pay attention to some areas like:

- **Marketing Differences:** Usually, the software will have some marketing material also. Check whether the software performs operations as mentioned in the marketing material. You must also ensure that the software is compared to the sales information and not the specification.

Language and Spelling:

Sometimes errors are created due to the poor language and vocabulary of programmers and writers. Thus, messages like '**If there are any discrepancies, please contact us immediately to ensure timely delivery of the products that you ordered'** may appear.

- ***Bad Media:*** Media is considered as the channel through which all the supporting items such as icons, images, sounds, and videos go with the software user interface. Thus it is essential to check that none of the supplied media are in bad condition

- ***What You See Is What You Get (WYSIWYG):*** Always check whether the user interface displayed is the one you have.
- ***Useful:*** A user interface has to be useful. The features of the product must be easy to use.

Accessibility Testing

- Accessibility testing is a technique used to ensure that the software product is accessible to people with disabilities such as visually challenged, physically challenged, and hearing defects. Accessibility testing is also known as testing for the disabled.

Accessibility Testing

- Accessibility testing is classified into four groups based on types of accessing difficulties and issues.

- ***Visual Impairments:*** People with visual impairments like blindness, restricted vision, color blindness, and so on can have the ease of working with the software products.
- People with this kind of disability (visual impairments) make use of the assistive technology software.
- Assistive technology through screen reading software helps the blind read the content. This technology simulates the human voice reading the text on computer screen or renders hard copy output into Braille. An example of such technology is Job Access With Speech (JAWS).

Accessibility Testing

- ***Motor Skills:*** Motor skills involve the skills pertaining to usage of keyboards and mouse. People with an inability to use the keyboard or mouse and to make fine movements on the software product come under this category.
- Hearing Impairments: Hearing impairments relate to people with a reduced or total loss of hearing.
- Cognitive Abilities: The cognitive abilities relate to reasoning and judging skills of an individual. People with reading disabilities, dyslexia, or memory loss fall under this category.

Accessibility Testing

- When performing accessibility testing, a tester must ensure that an accessibility checklist is used to certify the compliance related to accessibility.
- It provides information about what needs to be tested, how it should be tested, and the status of the software product with respect to various access related problems.

Accessibility Testing

Following is a simple test case for accessibility testing:

1. Ensure that all the functions are available with the keyboard, without having to use the mouse.

2. Ensure that information is visible when the display settings are set to high contrast modes.

3. Ensure that screen reading tools read the available texts as pictures, and images contain alternate texts associated with them.

4. Ensure that the product-defined keyboard actions do not affect keyboard shortcuts associated with accessibility.

Accessibility testing

- Two categories of tools used for accessibility testing.

(a) **Inspectors or Web Checkers:** This category enables the developer and the tester to know the exact information provided to an assistive technology. Tools like inspect object are used to obtain information given to a system on the assistive technology.

(b) **Assistive Technologies (AT):** This category is used by people with disability. Tools like screen readers, screen magnifiers, and so on are used to enhance accessibility. Testing an assistive technology is performed manually to have a better understanding of the interaction between AT and the product and documentation.

Questions