

Assignment 3: Based on Testing types and levels

### **1. Difference between Sanity and Smoke Testing?**

Smoke Testing is like a quick check to see if the basic functions of the software are working properly. It's done after a new build is released to make sure that the main features work and that the build is stable enough for further testing. If the software passes smoke testing, testers can proceed with more detailed testing.

Sanity Testing, on the other hand, is a more focused check that happens after receiving a software build with specific changes or bug fixes. It verifies that the new changes work as intended and that they haven't caused any new issues in the related parts of the software. Sanity testing is done to confirm that the changes made are effective before doing more extensive testing.

### **2. What do you test in Functional Testing?**

In **functional testing**, you test the software to ensure it behaves according to the requirements and specifications. Here are the key things you typically check:

1. **Features and Functions:** You verify that all features and functions of the application work as intended. This includes buttons, links, forms, and any other interactive elements.
2. **User Interface:** You check the layout, design, and usability of the user interface to make sure it's user-friendly and matches design specifications.
3. **Input Handling:** You test how the software handles different types of input, including valid data, invalid data, and edge cases. This helps ensure the system can handle various user scenarios without crashing.
4. **Data Processing:** You verify that the software processes data correctly, including calculations, data manipulation, and data storage.
5. **Error Handling:** You check how the software responds to errors. This includes making sure that error messages are displayed correctly and that the application doesn't crash.
6. **Integration with Other Systems:** If the software interacts with other systems or applications, you test those integrations to ensure data flows correctly between them.
7. **Security Features:** You check security functions, like user authentication and authorization, to make sure sensitive data is protected.

### **3. What Is Mean by Equivalence Partitioning?**

Equivalence Partitioning is a testing technique that helps make testing easier by reducing the number of test cases needed. The main idea is to split input data into groups, or "partitions," that the software treats the same way.

1. **Identify Input Conditions:** First, you look at the different inputs the software can take, like fields in a form or values in a function.
2. **Divide into Partitions:** For each input condition, you divide the possible inputs into different groups. Each group represents a set of values that should give similar results. For example, if a field accepts numbers between 1 and 100, you could have these groups:
  - Valid inputs: 1-100
  - Invalid inputs: Less than 1

- Invalid inputs: Greater than 100
3. **Select Test Cases:** Instead of testing every possible value, you pick one example from each group. This way, if one value from a group works correctly, others in that group are likely to work too.

#### **4. When to Do Smoke Testing?**

Smoke testing is done right after a new build or version of the software is released. Here are some situations when you should perform smoke testing:

1. **After a New Build:** Whenever a new version of the software is created, smoke testing checks if the basic functions work. This ensures that the build is stable enough for further testing.
2. **After Major Changes:** If significant changes have been made, like adding new features or fixing bugs, smoke testing should be done to confirm that these changes haven't broken anything important.
3. **Before Detailed Testing:** Before starting more thorough testing (like functional or regression testing), it's essential to do smoke testing. This helps to confirm that the software is ready for more in-depth checks.
4. **On Frequent Builds:** In a continuous integration environment, where builds happen often, smoke testing should be done regularly to quickly catch any major issues early.

#### **5. Why do we need to conduct end-to-end Testing?**

1. **Verify Workflow:** to ensures that all components of the application work together as expected. This includes checking user interactions, data flows, and system integrations.
2. **Catch Issues Early:** By testing the complete system, end-to-end testing helps identify issues that might not be caught when testing individual components. This means problems can be fixed before the software is released to users.
3. **Test Real User Scenarios:** End-to-end testing simulates how real users will use the software. It checks if the application behaves correctly in real-world conditions, which helps ensure a good user experience.
4. **Check Data Integrity:** It verifies that data is correctly passed between different parts of the application and remains accurate throughout the process.
5. **Improve Quality Assurance:** Conducting end-to-end tests increases the overall quality of the software. It helps build confidence that the system works as intended, leading to fewer issues in production.

#### **6. What is RTM?**

RTM stands for Requirement Traceability Matrix. It is a document used in software development to track and manage the requirements of a project.

1. **Tracks Requirements:** RTM helps keep track of all the requirements from the beginning of a project to the end. It shows which requirements have been met and which still need to be addressed.

2. **Links to Test Cases:** Each requirement in the RTM is linked to specific test cases. This means that for every requirement, there are corresponding tests that ensure it has been implemented correctly.
3. **Ensures Coverage:** By using RTM, teams can ensure that all requirements are tested. This helps prevent any important features from being overlooked and ensures that the final product meets user needs.
4. **Facilitates Change Management:** If requirements change during the project, the RTM helps manage those changes by updating the linked test cases and tracking the impact on the project.
5. **Improves Communication:** RTM serves as a communication tool among stakeholders, including developers, testers, and project managers. It keeps everyone on the same page regarding what needs to be done.

## **7. What is GUI testing?**

GUI Testing, or Graphical User Interface Testing, is a type of testing that checks the part of the software that users see and interact with. The main goal is to make sure everything looks good and works properly. Here are the main points about GUI testing:

1. **Looks Good:** Testers check that the layout, colors, fonts, and images match what was designed. The software should be visually appealing.
2. **Works Right:** They verify that all buttons, links, menus, and forms function correctly. For example, clicking a button should do what it's supposed to do.
3. **Easy to Use:** This testing checks if the software is user-friendly. Testers look at how easy it is for users to navigate and complete tasks.
4. **Compatible:** GUI testing makes sure the interface works well on different devices, browsers, and screen sizes. Users should have a consistent experience no matter how they access the software.
5. **Handles Errors Well:** Testers check that error messages show up correctly when users make mistakes, like entering wrong information in a form.

## **8. Difference between Retesting and Regression Testing? Explain with Real life Example.**

Retesting and regression testing are two important testing types.

### **1. Retesting:**

- Retesting is done to check if specific bugs or issues have been fixed after they were reported.
- It focuses only on the particular defect that was fixed.
- Imagine we are using a mobile app, and you find a bug where the "Save" button doesn't work. The developers fix the issue, and you then run the app again to specifically test the "Save" button to confirm that it now works correctly.

### **2. Regression Testing:**

- Regression testing is done to ensure that recent changes or fixes in the software have not affected other parts of the application.

- It checks the entire application or a large part of it, not just the fixed bug.
- Using the same mobile app, after fixing the "Save" button, the developers also add a new feature, like a "Share" button. Regression testing checks that both the "Save" and "Share" buttons work correctly and that nothing else in the app is broken because of the changes made.

## **9. Write test cases on dot (.)**

Description: Check if the application accepts a single dot as valid input.

Check if the application accepts multiple dots in the input.

- Check if the application accepts a dot at the beginning of the input.
- Check if the application accepts a dot at the end of the input.
- Check if the application accepts a dot in the middle of the input.
- Check if the application handles dots incorrectly used in the input (e.g., two dots in a row).
- Input: "test..case"
- Expected Result: The application should show an error message indicating invalid input.

### **2. Test Case: Dot in Numeric Input**

- Description: Check if the application accepts a dot in a numeric input (e.g., for decimal numbers).
- Input: "3.14"
- Expected Result: The application should accept the input as a valid decimal number.

## **10. Write Uses of a pen except writing.**

1. Bookmark: You can use a pen to mark your place in a book or a notebook.
2. Paperweight: If you have loose papers, a pen can hold them down on your desk or table.
3. Stylus: A pen can be used as a makeshift stylus for touchscreens, especially if the tip is fine enough.
4. Drawing Tool: You can use a pen for sketching or drawing when you want to create art.
5. Bubble Popper: The pointy end of a pen can be used to pop bubble wrap or other inflatable items.
6. Measuring Tool: You can use a pen to measure small distances or sizes by using it as a straight edge.
7. Hair Tie: In a pinch, a pen can be used to hold your hair back if you don't have a hair tie.
8. Gift Wrapping Tool: You can use a pen to create designs or write messages on gift wrap.
9. Cleaning Tool: The end of a pen can be used to clean small spaces, like getting dirt out from between keys on a keyboard.
10. Lock Pick: In a non-destructive way, you can sometimes use a pen to unlock a simple lock if you're locked out.

11. Scratch Paper: The back of a pen cap can be used as a surface to jot down quick notes or ideas when you don't have paper.
12. Screwdriver: In a pinch, a pen can help turn small screws if you carefully use the end to grip and twist.
13. Stress Relief: You can use a pen to click or spin when you're stressed, providing a simple fidget tool.
14. Whistle: If you blow through a pen (after removing the ink), it can create a whistle sound for getting someone's attention.

## **11. How to categories the testing .Write name of all testing types?**

1. Based on Level of Testing:
  - Unit Testing: Testing individual components or functions of the software.
  - Integration Testing: Testing the interaction between integrated components or systems.
  - System Testing: Testing the complete and integrated software as a whole.
  - Acceptance Testing: Testing to determine if the software meets business requirements and is ready for use.
2. Based on Type of Testing:
  - Functional Testing: Testing the software against functional requirements.
  - Non-Functional Testing: Testing aspects not related to specific functions, like performance and security.
3. Based on Approach:
  - Manual Testing: Testing performed by humans without automation tools.
  - Automated Testing: Testing using scripts and tools to automate test execution.
4. Based on Execution Time:
  - Static Testing: Testing the software without executing it, like reviews and inspections.
  - Dynamic Testing: Testing by executing the software and observing its behavior.
5. Based on Scope:
  - Smoke Testing: A quick check to see if the basic features of the software work after a build.
  - Sanity Testing: A focused check to ensure specific functionalities work after changes.
  - Regression Testing: Testing to ensure that recent changes haven't broken existing features.
6. Based on Usage:
  - Alpha Testing: Testing done by developers or internal staff before releasing to external users.
  - Beta Testing: Testing done by real users in a real environment before the final release.

## 1. Unit Testing

- **Description:** Tests individual components or functions of the software in isolation to ensure they work correctly.
- **Subtypes:**
  - **Positive Testing:** Verifies that the unit works with valid inputs.
    - **Example:** Testing a function that calculates the square root to ensure `sqrt(9)` returns 3.
  - **Negative Testing:** Tests how the unit handles invalid inputs or edge cases.
    - **Example:** Checking that `sqrt(-1)` returns an error or handles it gracefully.
- **Example:** A function that checks if a number is prime can be unit tested with inputs like 3 (expect true) and 4 (expect false).

## 2. Integration Testing

- **Description:** Checks how different units or modules interact with each other to ensure they function together correctly.
- **Subtypes:**
  - **Big Bang Integration Testing:** All components are integrated at once and then tested together.
    - **Example:** An entire e-commerce application is tested after all its modules (user authentication, payment processing, etc.) are integrated.
  - **Incremental Integration Testing:** Components are integrated and tested one at a time or in small groups.
    - **Example:** First testing the user authentication module with the database module, then adding the payment processing module.
- **Example:** Testing the interaction between the front-end user interface and back-end database when a user submits a form.

## 3. System Testing

- **Description:** Evaluates the complete and integrated software system to ensure it meets specified requirements.
- **Subtypes:**
  - **Functional Testing:** Verifies that the software behaves according to functional requirements.
    - **Example:** Testing all features of a new mobile app to ensure they work as described in the requirements document.
  - **Non-Functional Testing:** Checks aspects like performance, usability, and security.
    - **Example:** Stress testing an application to see how it performs under heavy load.
- **Example:** Testing the overall flow of an online shopping platform, including product selection, cart functionality, and checkout process.

## 4. Acceptance Testing

- **Description:** The final level of testing to confirm that the software meets user needs and is ready for release.

- **Subtypes:**
  - **User Acceptance Testing (UAT):** Conducted by end users to ensure the software meets their requirements.
    - **Example:** A group of end users testing a new feature in a software application to see if it meets their needs before the official launch.
  - **Operational Acceptance Testing:** Focuses on operational readiness, including backup and recovery procedures.
    - **Example:** Ensuring that an application can recover data after a crash.
- **Example:** Before releasing a new version of a project management tool, real users test it to confirm that new features work and improve their workflow.

## 12. Difference between HTTP and HTTPS.

HTTP	HTTPS
Not secure	Secure, encrypted using SSL/TLS
No encryption; data is sent in plain text	Encrypts data during transfer, making it unreadable to attackers
Uses port 80	Uses port 443
Begins with "http://"	Begins with "https://"
Suitable for non-sensitive data	Essential for sensitive data, like online banking or e-commerce
Lower ranking in search engines	Higher ranking; preferred by search engines
Generally faster for non-sensitive pages	Slightly slower due to encryption overhead, but minimal impact
Less trustworthy; users may see warnings	More trustworthy; users see a padlock icon indicating security

## 13. What is FTP and why we use it.?

1. **Security:** HTTP is not secure, meaning that data is sent in plain text and can be read by anyone who intercepts it. HTTPS is secure because it encrypts the data, making it hard for others to read.
2. **Data Encryption:** With HTTP, your information travels openly, while HTTPS protects your data by scrambling it during transfer.
3. **Port Numbers:** HTTP usually works on port 80, while HTTPS uses port 443.
4. **URL Appearance:** Websites that use HTTP start with "http://", and those that use HTTPS start with "https://". The "s" at the end stands for "secure."
5. **When to Use:** HTTP is fine for regular web pages, but HTTPS is necessary for any site that handles sensitive information, like online shopping or banking.
6. **Trust:** Users feel safer on HTTPS sites because they see a padlock icon in the address bar, showing that the connection is secure.
7. **Search Engine Ranking:** HTTPS sites often rank higher in search results because search engines prefer secure connections.

8. Speed: HTTPS may be slightly slower due to the extra security measures, but the difference is usually minimal.

#### **14. Full form of XAMPP,WAMP,LAMP,MAMP.**

1. XAMPP:

- Full Form: Cross-Platform, Apache, MySQL, PHP, and Perl.

2. WAMP:

- Full Form: Windows, Apache, MySQL, and PHP.

3. LAMP:

- Full Form: Linux, Apache, MySQL, and PHP/Python/Perl.

4. MAMP:

- Full Form: MacOS, Apache, MySQL, and PHP.

#### **15. Latest version of difference browsers.**

1. Google Chrome:

- Latest Version: Chrome 118.0.5993.90
- Release Date: November 2024

2. Mozilla Firefox:

- Latest Version: Firefox 119.0
- Release Date: November 2024

3. Microsoft Edge:

- Latest Version: Edge 118.0.2045.43
- Release Date: November 2024

4. Apple Safari:

- Latest Version: Safari 17.0
- Release Date: September 2024

5. Opera:

- Latest Version: Opera 102.0.4880.61
- Release Date: November 2024

6. Brave:

- Latest Version: Brave 1.50.113
- Release Date: November 2024



## 16. Which types of bugs find in Black Box testing?

Functional Bugs: These occur when the software doesn't behave as expected according to the requirements.

- *Example:* A login button that doesn't work when the correct credentials are entered.
- UI Bugs: Issues with the user interface, like alignment problems, font errors, or color mismatches.
- *Example:* A button that is misaligned with the text on the screen.
- Performance Bugs: Problems related to speed and responsiveness of the application.
- *Example:* A webpage that takes too long to load or crashes under heavy traffic.
- Usability Bugs: Issues that affect the user experience, making the software difficult to use.
- *Example:* Confusing navigation that makes it hard for users to find features.
- Compatibility Bugs: These occur when the software doesn't work well across different browsers or devices.

## 17. What do you mean by Responsiveness ?

Responsiveness in web design refers to how well a website adapts to different screen sizes and devices. A responsive design ensures that the layout, images, and text resize and rearrange to provide a good user experience, whether on a desktop, tablet, or smartphone.

- Key Points:
  - Flexible Layout: The design adjusts fluidly to fit any screen size.
  - Adaptable Images: Images scale properly without losing quality or becoming distorted.
  - User-Friendly Navigation: Menus and buttons are easy to click and use on all devices.

### **18. What is Performance Testing? Explain the difference between Load Testing , Stress Testing and Volume Testing.**

Performance testing checks how well a software application performs under various conditions. It aims to ensure the application is fast, responsive, and stable, especially under high loads. Here's a breakdown of three types of performance testing:

#### **1. Load Testing:**

- Description: Tests how the application behaves under expected load conditions, like a certain number of users.
- Example: Checking if an online store can handle 1000 users at the same time without slowing down.

#### **2. Stress Testing:**

- Description: Tests the application's limits by pushing it beyond normal load to see how it handles high traffic or data.
- Example: Simulating 5000 users on the online store to see if it crashes or fails gracefully.

#### **3. Volume Testing:**

- Description: Tests how the application handles large volumes of data.
- Example: Uploading a huge database to see if the application can process it without issues.