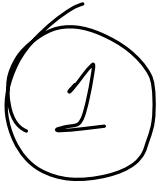
Advanced Topics

SE401: Software Quality Assurance and Testing



Outline

- Web Testing
- Mobile Testing

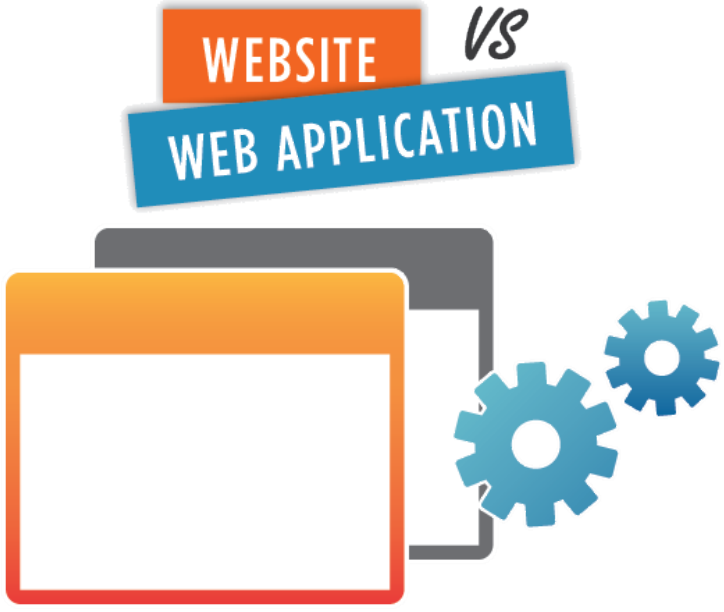
Web Testing

Outline

- Web-site vs. Web Application
- Testing Web Applications
- Web Testing Types
- Test Automation

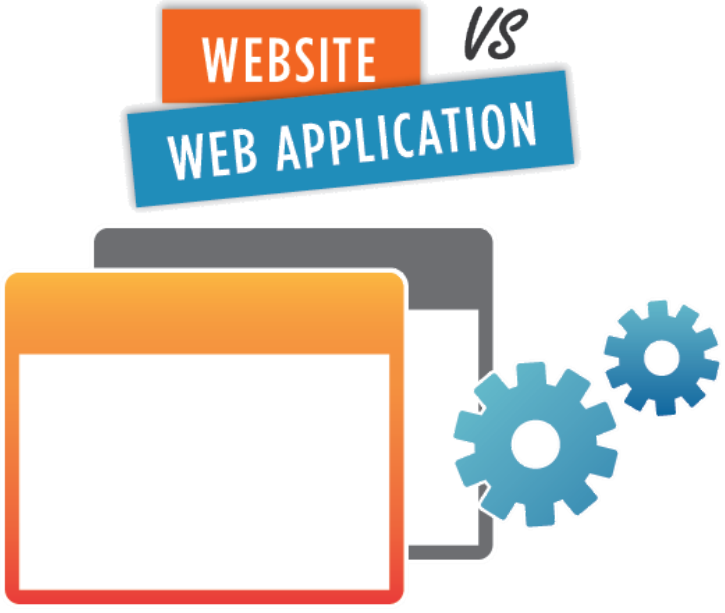
Website vs. Web Application

- **A website** is a collection of related web pages, including multimedia content, typically identified with a common domain name, and published on at least one web server. Notable examples are wikipedia.org.
- **So** .. A website is informational



Website vs. Web Application

- **A web application:** is a client–server computer program which the client (including the user interface and client-side logic) runs in a web browser. Common web applications include webmail, online retail sales, and online auction.
- **So** .. a web application is interactive



So ..

Website

Static pages:

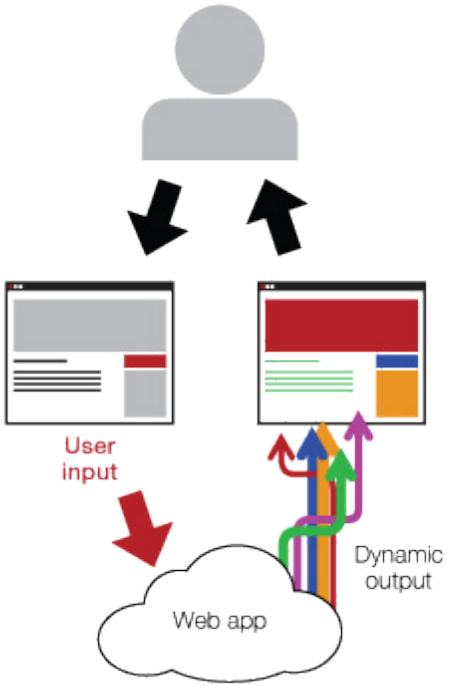
Every user sees the same information each time



Web App

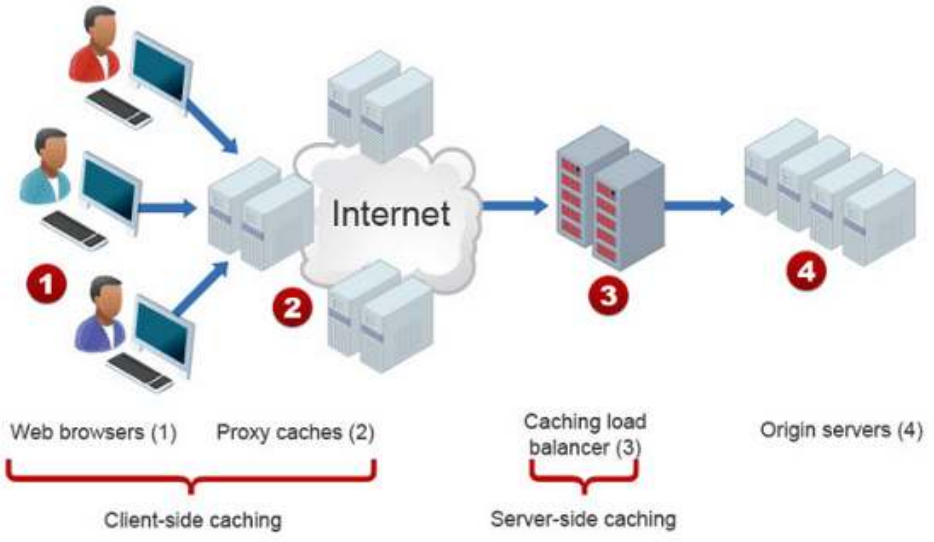
Dynamic pages:

Every user sees the different information depending on their input



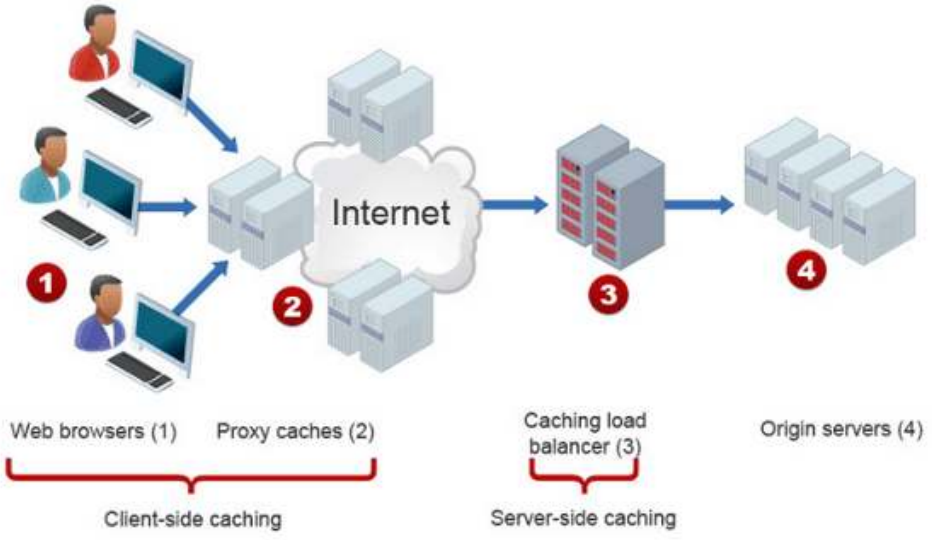
What is Web Testing

- Web testing is a software testing practice to test the **websites** or **web applications** for potential bugs.
- A web-based system needs to be checked **completely** from **end-to-end** before it goes live for end users.



What is Web Testing

- It is performed **before** making the web application alive.
- By performing web testing, we can make sure that the website or web application is functioning properly and can be accepted by real-time users.



Why Web Testing is Challenging?

- Web based systems and applications interoperate with many different:
  - **Operating** systems
  - **Browsers**
  - **Hardware** platforms
  - **Communications** protocols
- The search for errors represents a significant challenge
- New technologies make web applications rich, **but** they become more difficult to test.

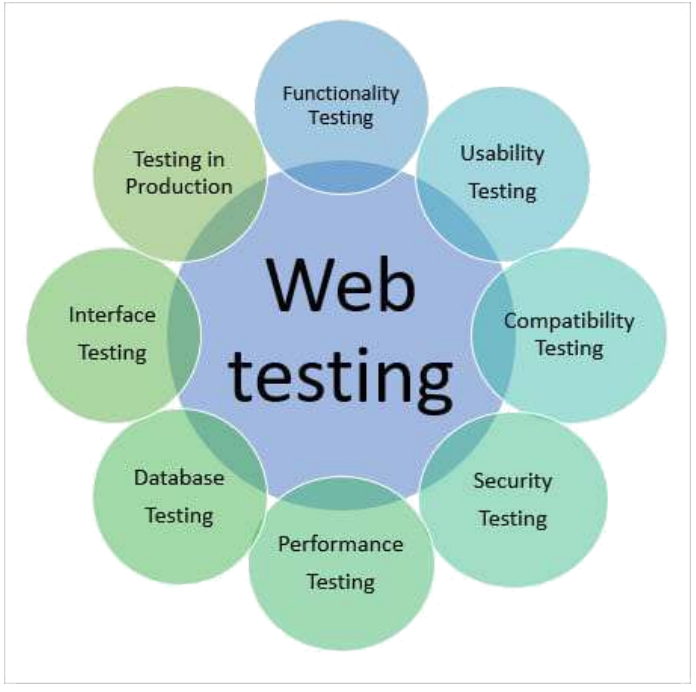
Why's web testing challanging?

- Different OS.
- Different HW.
- Different Browsers.
- Communication Protocols.



Web Testing Types

- Some or all the following testing types may be performed depending on the web testing requirements:
  1. **Functionality** Testing
  2. **Usability** testing
  3. **Interface** testing
  4. **Compatibility** testing
  5. **Performance** testing
  6. **Security** testing



1. Functionality Testing

- Testing the **features** and operational behavior of a web application to ensure they correspond to its **specifications**.

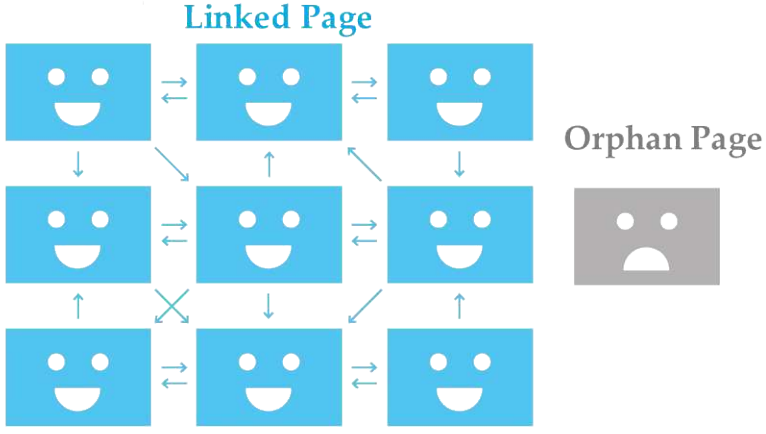
does a link links me to the other page?

- Test for – all the **links** in web pages, **database** connection, **forms** used for submitting or getting information from the user in the web pages, **Cookie** testing, and **HTML and CSS files** etc.



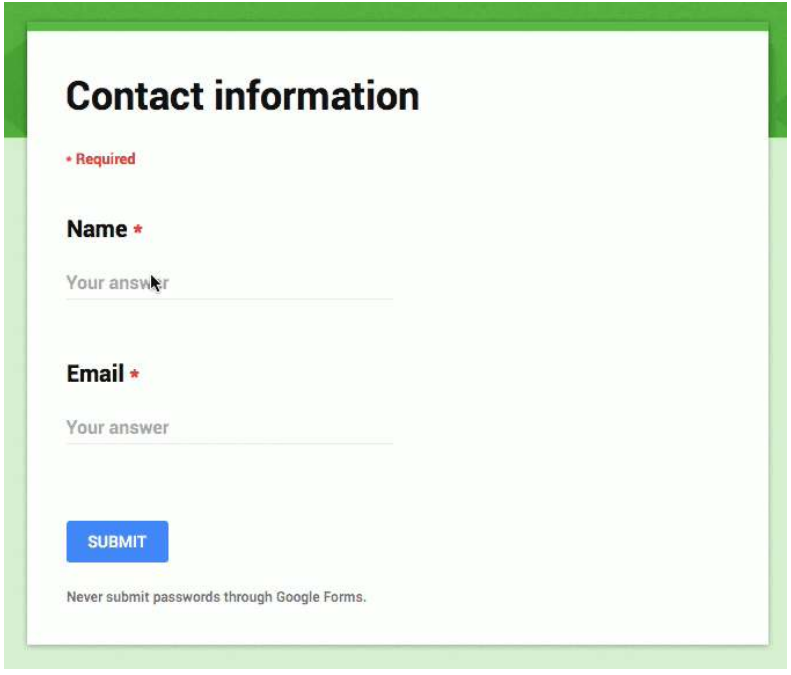
1. Functionality Testing

- **1.1 Check all the links:**
  - Test the external links from all the pages to a specific domain under test.
  - Test all internal links between web application pages
  - Test links used to send email to admin or other users from web pages.
  - Test to check if there are any orphan pages.
  - Finally, link checking includes, check for broken links in all above-mentioned links.



1. Functionality Testing

- **1.2 Test forms on all pages:** Forms are part of any website that is used for **receiving** information from users and to **interact** with them.
- So what should be checked in these forms?
  - First, check all the validations on each field.
  - Check for default values of the fields.
  - Wrong inputs to the fields in the forms.



1. Functionality Testing

- **1.3 Test Cookies:** cookies are small files that are stored on a user's computer. They hold data specific to a particular client and website, and can be accessed either by the web server or the client computer.
  - Test the application by enabling or disabling cookies in a browser
  - Test if cookies are encrypted before they are written to the user machine
  - Test if cookies are deleted when they are not needed anymore.



1. Functionality Testing

- **1.4 Test HTML and CSS:**
  - Checking for **syntax errors**
  - Check if the site is **crawlable** by different search engines.
  - Readable **color scheme** and **fonts**
  - **Standards** compliance (i.e. such as W3C and ISO)



1. Functionality Testing

- **1.5 Database Testing:**
  - Data validity testing: test if any errors are shown while executing queries
  - Data integrity testing: check for errors while you edit, delete, modify data in database
  - Check response time of queries.
  - Check if all the database queries are executing correctly, data is retrieved and updated correctly.
  - Check if data retrieved from database is shown accurately in the web application .



1. Functionality Testing

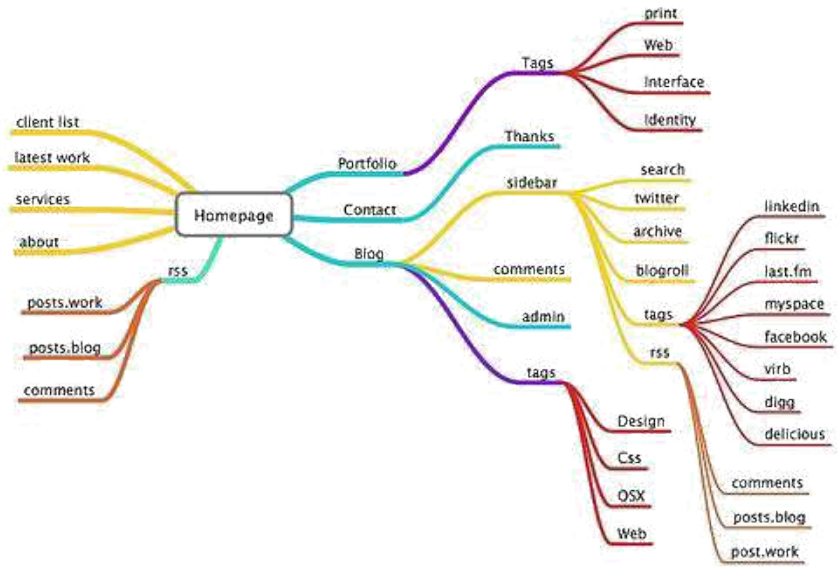
- **Examples of Functionality Test Scenarios:**
  - Test all the mandatory fields should be validated.
  - Test the system should not display the error message for optional fields.
  - Test the numeric fields should not accept the alphabets and proper error message should display.
  - Test the pop-up message ("This field is limited to 500 characters") should display if the data reaches the maximum size of the field.
  - Test that a confirmation message should display for update and delete operations.
  - Test all input fields for special characters.
  - Test the functionality of the buttons available, … etc.

2. Usability Testing

- **Usability testing** is the process by which the human-computer interaction characteristics of a system are measured, and weaknesses are identified for correction.
- In usability testing:
  - Check if the application is **user-friendly** (general appearance and interaction)
  - Check the **flow** of the system and that it is easy to understand and learn by the user
  - Basically, system **navigation** and **content** are checked in usability testing

2. Usability Testing

- **2.1 Navigation:**
  - Navigation means how a user **surfs** the web pages, different controls like buttons, boxes or how the user uses the links on the pages to surf different pages.
  - **Menus**, **buttons** or **links** to different pages on your site should be **easily visible** and **consistent** on all webpages
  - The web application should be **easy** to use and **navigate**



2. Usability Testing

- **2.2 Content:**
  - Content should be **logical** and **easy** to **understand**. Check for spelling errors. Usage of dark colors annoys the users and should not be used in the site theme.
  - You can follow some **standard colors** that are used for web page and content building.
  - Content should be **meaningful**. All the anchor text links should be working properly. Images should be placed properly with proper sizes.
  - These are some of the basic important **standards** that should be followed in web development.

2. Usability Testing

- **Examples of Usability Test Scenarios:**
  - All fonts should be same as per the requirements.
  - All the text should be properly aligned.
  - Tool tip text should be there for every field.
  - All the buttons should be in a standard format and size.
  - Home link should be there on every single page.
  - Confirmation message should be displayed for any kind of interaction.
  - Check the site on different resolutions (640 x 480, 600x800 etc.?)
  - Check the end user can run the system without frustration.
  - Scroll bar should appear only if required.
  - All fields and buttons should be accessible by keyboard shortcuts and the user should be able to perform all operations by using keyboard… etc.

3. Interface Testing

- The main **layers** in a web application are:
  - Web server
  - Application server
  - Database server
- Check if all **interactions** between these layers are executed properly
- Interactions between different pages in the applications are also checked

4. Compatibility Testing

- Compatibility testing is used to determine if your software is compatible with other elements of a system with which it should operate, e.g. **Browsers**, **Operating Systems**, or **hardware**.
- The purpose of Compatibility testing is to evaluate **how well software performs** under different browsers, operating systems, hardware or software.

4. Compatibility Testing

- **Examples of Compatibility Test Scenarios:**
  - Test the web application in different browsers (IE, Firefox, Chrome, Safari and Opera) and ensure the website is displaying properly.
  - Test the HTML version being used is compatible with appropriate browser versions.
  - Test the images display correctly in different browsers.
  - Test the fonts are usable in different browsers.
  - Test the java script code is usable in different browsers.
  - Test the Animated GIF's across different browsers.
  - Etc..

5. Security Testing

- Security Testing is vital for web applications that store sensitive customer information like credit cards, ids, passwords, etc.
- Testing activities will include:
  - Test unauthorized access to secure pages should not be permitted
  - Restricted files should not be downloadable without appropriate access
  - Check sessions are automatically killed after prolonged user inactivity
  - On use of SSL certificates, website should re-direct to encrypted SSL pages.

5. Security Testing

- **Examples of Security Test Scenarios:**
  - Verify the web page which contains important data like password, credit card numbers, secret answers for security question etc should be submitted via HTTPS (SSL).
  - Verify the important information like password, credit card numbers etc should display in encrypted format.
  - Verify password rules are implemented on all authentication pages like Registration, forgot password, change password.
  - Verify if the user is logged out from the system or user session was expired, the user should not be able to navigate the site.
  - Verify the user account gets locked out if the user is entering the wrong password several times.
  - Verify the user roles and their rights. For Example The requestor should not be able to access the admin page.
  - Etc..

6. Performance Testing

- This will ensure your site works under all loads. Testing activities will include but not limited to:
  - Website application response times at different connection speeds
  - Load test your web application to determine its behavior under normal and peak loads
  - Stress test your web site to determine its break point when pushed to beyond normal loads at peak time.
  - Test if a crash occurs due to peak load, how does the site recover from such an event

6. Performance Testing

- **Examples of Performance Test Scenarios:**
  - To determine the performance, stability and scalability of an application under different load conditions.
  - To determine if the current architecture can support the application at peak user levels.
  - To determine which configuration provides the best performance level.
  - To identify application and infrastructure bottlenecks.
  - To determine if the new version of the software adversely had an impact on response time.
  - To evaluate product and/or hardware to determine if it can handle projected load volumes.
  - Etc..

Website Test Automation

- Automation testing helps to relieve the bulk of tedious and repetitive human tasks.
- With the right test automation framework or tool, QAs can automate repetitive browser actions like interacting with web elements or filling out long HTML forms.
- Automating tests help teams save time and effort, get faster and accurate results, increase software efficiency, and focus on innovation.

Benefits of Website Test Automation

- Faster feedback
- Improved test efficiency
- Reduced expenses
- Reusability of test cases
- Faster time to market

Selenium

Selenium is open source testing tool for web applications. Any web application can be testing on web browsers with this automation testing tool.

Selenium tool can test only web application, it is not useful for mobile or desktop applications.

- Test scripts can be written in *Java, Python, C#, PHP, Ruby, Perl & .Net*
- Tests can be carried out in *Windows, Mac or Linux*
- Tests can be carried out using *Mozilla Firefox, Internet Explorer, Google Chrome, Safari or Opera*
- It can be integrated with tools like *JUnit* for managing test cases and generating reports

Setup Selenium

- Eclipse IDE (Users can use other IDEs too)
- Selenium Client and language WebDiver bindings
- Configuring Selenium WebDriver with Eclipse

Summary

- Web-site vs. Web Application
- Testing Web Applications
- Web Testing Types
- Test Automation

Mobile Testing

Outline

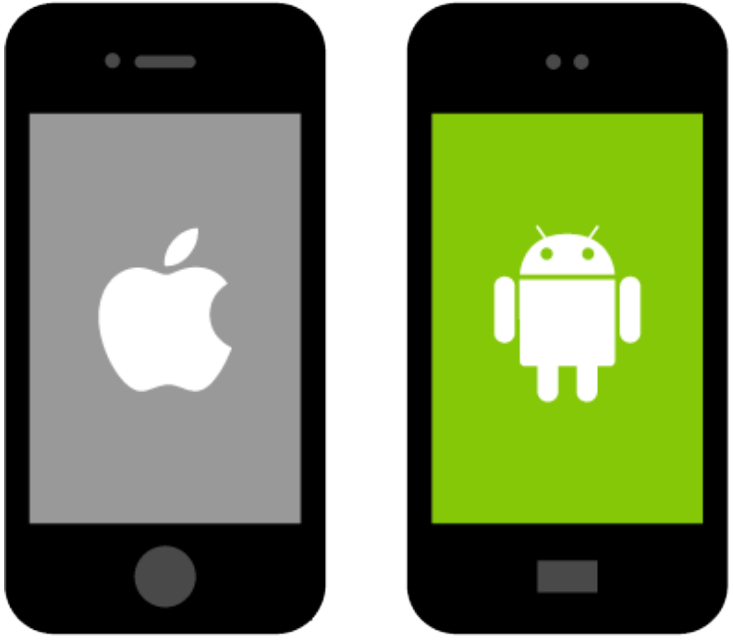
- What is a Mobile Application?
- Mobile Testing vs. Mobile Application Testing
- Why Perform Mobile Application Testing
- Challenges in Mobile Application Testing
- Types of Mobile Application Testing

Mobile Application

- A **mobile application** is a software application developed specifically for use on small, wireless, handheld mobile devices such as smart-phones and tablets.
- Mobile apps are available through application distribution platforms which are typically operated by the owner of the mobile operating system, such as the App Store (iOS) or Google Play Store.



Main Mobile Phone Platforms



Mobile Application

- **Types of Mobile Applications:**
  - Native Applications: example: App Store, Settings.
  - Mobile Web Applications
  - Hybrid applications Netflix.



Mobile Application

- **Native Apps:**
  - Are installed through different mobile platforms stores (Google Play Store or Apple's app store).
  - Are Platform specific
  - Offers fast, most reliable and most responsive experience to the users.



Mobile Application

- **Mobile Web Apps:**
  - Are server-side apps to access website/s on mobile using different browsers like Chrome, Firefox by connecting to a mobile network or wireless network like Wi-Fi.
  - Developing web apps is simple and quicker than Native apps
  - Slower and less intuitive comparted to Native apps.



Mobile Application

- **Hybrid Apps:**
  - Are combinations of native app and web app. (You install it like a native app, but it's a web app on the inside).
  - They run on devices or offline and are written using web technologies like HTML5 and CSS.



Mobile vs Mobile Application Testing

- **Mobile Testing:**
  - Testing the mobile device (the **hardware**) including the internal processors, internal hardware, screen sizes, resolution, space or memory, camera, radio, Bluetooth, WIFI etc.
- **Mobile Application Testing:**
  - Testing the applications (**software**) that work on mobile devices and their functionality.



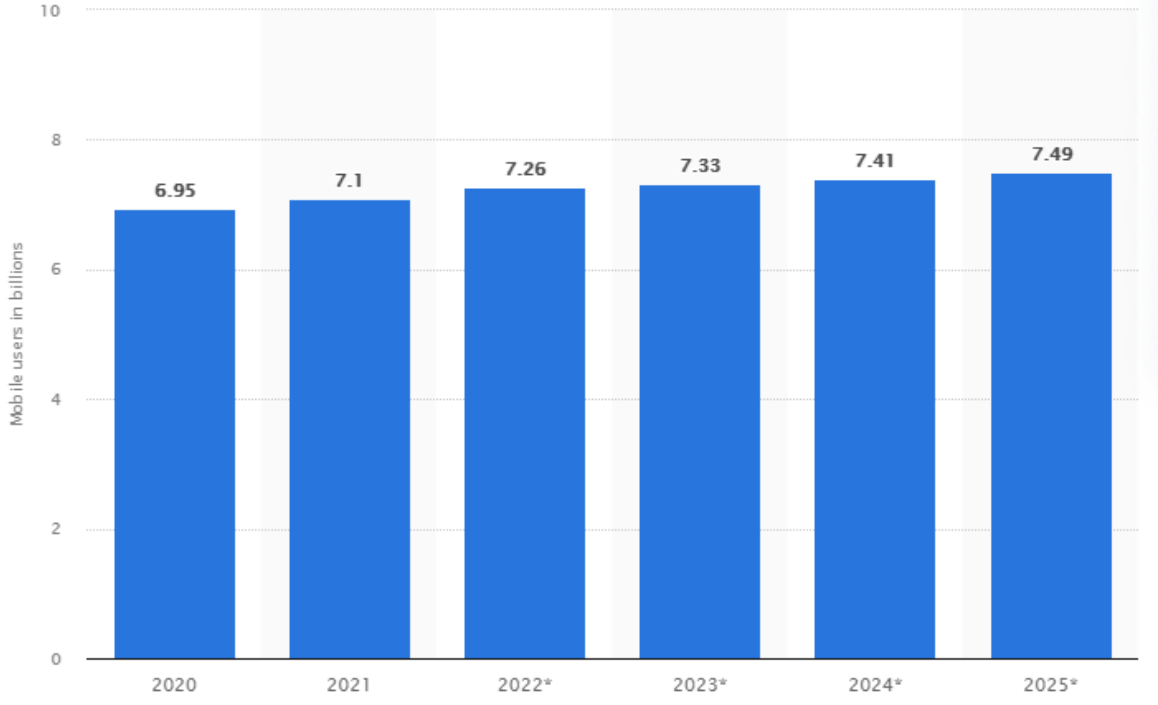
Mobile Application Testing



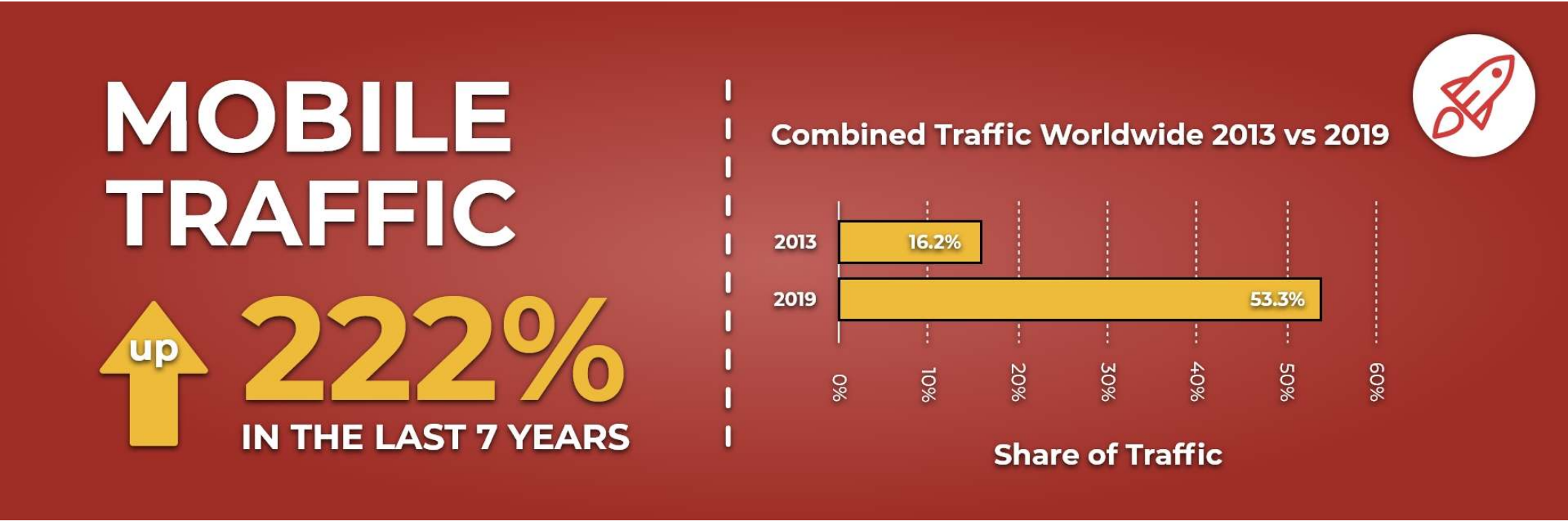
- **Mobile application testing** is a process by which application software developed for handheld mobile devices is tested for its functionality, usability and consistency.
- Mobile application testing can be an **automated** or **manual** type of testing.

Why Testing Mobile Apps is Important

Forecast number of mobile users worldwide from 2020 to 2025



Why Testing Mobile Apps is Important



Challenges in Mobile App testing

- Issues that make testing mobile applications more challenging:
  - Variety of mobile devices available in the market (i.e. HTC, iPhone, Samsung, etc..)
  - Diverse mobile operating platforms used (iOS, Android)
  - Different network service providers
  - Different input methods
  - Hardware compatibility

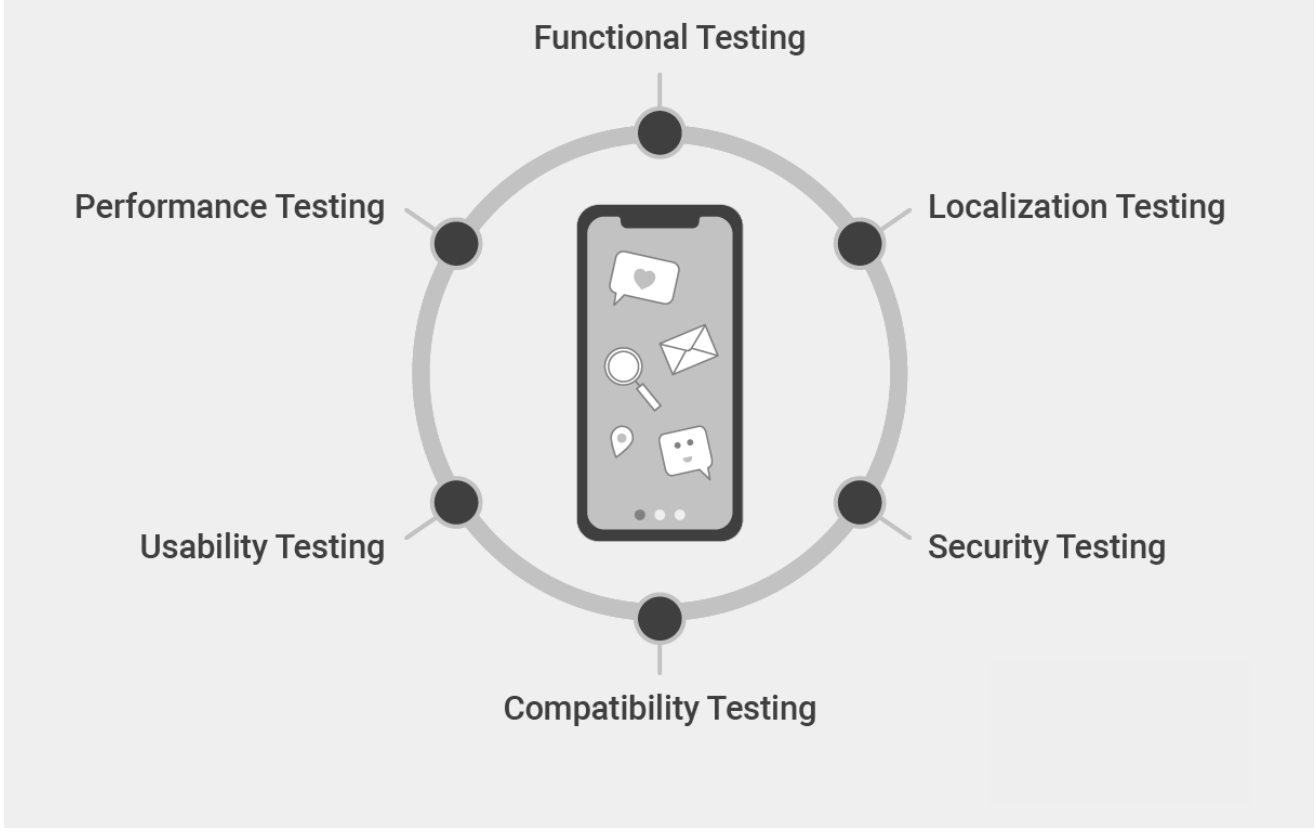


Mobile Environment Test Design Considerations

Area	Comment
Connectivity	Device provisioning Network speed Network latency Network availability in remote areas Service reliability
Diversity Devices	Numerous web browsers to test Multiple versions of runtimes for Java or other languages
Device Donstraints	Limited memory or processor Small screen size Multiple operating systems Multitasking capabilities Data cache sizes

Area	Comment
Input Devices	Touch screens Stylus Mouse Buttons Rollers
Installation and Maintenance	Installing and uninstalling Patching Upgrading

Major Types of Mobile Apps testing



Functional Testing

- Consists of
  - testing user interactions
  - testing the transactions
- Relevant factors:
  - Type of application (banking, gaming, social or business)
  - Target audience type (consumer, enterprise, education)
  - Distribution channel (e.g. Apple App Store, Google play, direct distribution)

Functional Testing Scenarios (1)

- Mandatory fields are working as required
- Mandatory fields are displayed distinctively from the non-mandatory fields
- The application works as per as requirement whenever the application starts/stops
- The application goes into minimized mode whenever there is an incoming phone call
- The phone is able to store, process and receive SMS whenever the app is running
- The device is able to perform required multitasking requirements whenever it is necessary to do so

Functional Testing Scenarios (2)

- The application allows necessary social network options such as sharing, posting and navigation etc.
- The application supports any payment gateway transaction such as Visa, Mastercard, Paypal etc.
- The page scrolling scenarios are being enabled in the application as necessary
- The navigation between relevant modules in the application are as per the requirement
- The truncation errors are absolutely to an affordable limit
- The user receives an appropriate error message like “Network error. Please try after some time” whenever there is any network error

Functional Testing Scenarios (3)

- Allows other applications to perform satisfactorily, and it does not eat into the memory of the other applications
- Resumes at the last operation in case of a hard reboot or system crash
- Installation can be done smoothly provided the user has the necessary resources and it does not lead to any significant errors
- Performs auto start facility according to the requirements
- Performs according to the requirement in all versions of Mobile that is 2g, 3g,4g and 5g
- Regression Testing to uncover new software bugs in existing areas of a system after changes have been made to them.
- Provides an available user guide for those who are not familiar to the app

Performance Testing

- Fundamental objective is to ensure that the application performs acceptably under certain performance requirements such as access by a huge number of users or the removal of a key infrastructure part like a database server.

Performance Testing Scenarios (1)

- Performs as per the requirement under different load conditions
- The current network coverage is able to support the application at peak, average and minimum user levels
- The existing client-server configuration setup provides the required optimum performance level
- Identify the various application and infrastructure bottlenecks which prevent the application to perform at the required acceptability levels
- Validate whether the response time of the application is as per as the requirements

Performance Testing Scenarios (2)

- Evaluate product and/or hardware to determine if it can handle projected load volumes
- Evaluate whether the battery life can support the application to perform under projected load volumes
- Validate application performance when network is changed to WIFI from 3G/4G/5G or vice versa
- Validate that the battery consumption, memory leaks, resources like GPS, Camera performance is well within required guidelines

Performance Testing Scenarios (3)

- Validate the application longevity whenever the user load is rigorous
- Validate the network performance while moving around with the device
- Validate the application performance when only intermittent phases of connectivity is required

Security Testing

- The fundamental objective of security testing is to ensure that the application's data and networking security requirements are met as per guidelines

Security Testing Scenarios (1)

- Able to withstand any brute force attack which is an automated process of trial and error used to guess a person's username, password or credit-card number
- Not permitting an attacker to access sensitive content or functionality without proper authentication
- Has a strong password protection system and it does not permit an attacker to obtain, change or recover another user's password
- Does not suffer from insufficient session expiration
- Identify the dynamic dependencies and take measures to prevent any attacker for accessing these vulnerabilities

Security Testing Scenarios (2)

- Prevent from SQL injection related attacks
- Identify and recover from any unmanaged code scenarios
- Certificates are validated, does the application implement Certificate Pinning or not
- Protect the application and the network from the denial of service attacks
- Analyze the data storage and data validation requirements
- Enable the session management for preventing unauthorized users to access unsolicited information

Security Testing Scenarios (3)

- Check if any cryptography code is broken and ensure that it is repaired
- Validate whether the business logic implementation is secured and not vulnerable to any attack from outside
- Analyze file system interactions, determine any vulnerability and correct these problems
- Validate the protocol handlers for example trying to reconfigure the default landing page for the application using a malicious iframe
- Protect against malicious client side injections
- Protect against malicious runtime injections

Security Testing Scenarios (4)

- Investigate file caching and prevent any malicious possibilities from the same
- Prevent from insecure data storage in the keyboard cache of the applications
- Investigate cookies and preventing any malicious deeds from the cookies
- Provide regular audits for data protection analysis
- Investigate custom created files and preventing any malicious deeds from the custom created files
- Prevent from buffer overflows and memory corruption cases
- Analyze different data streams and preventing any vulnerabilities from these

Usability Testing

- The usability testing process of the Mobile application is performed to have a quick and easy step application with less functionality than a slow and difficult application with many features
- The main objective is to ensure that we end up having an easy-to-use, intuitive and similar to industry-accepted interfaces which are widely used

Usability Testing Scenarios (1)

- Buttons should have the required size and be suitable to big fingers
- Buttons are placed in the same section of the screen to avoid confusion to the end users
- Icons are natural and consistent with the application
- Buttons, which have the same function should also have the same color
- Validation for the tapping zoom-in and zoom-out facilities should be enabled
- Keyboard input can be minimized in an appropriate manner

Usability Testing Scenarios (2)

- Application provides a method for going back or undoing an action, on touching the wrong item, within an acceptable duration
- Contextual menus are not overloaded because it has to be used quickly
- Text is kept simple and clear to be visible to the users
- Short sentences and paragraphs are readable to the end users
- Font size is big enough to be readable and not too big or too small
- Application prompts the user whenever the user starts downloading a large amount of data which may be not conducive for the application performance

Usability Testing Scenarios (3)

- Closing of the application is performed from different states and verify if it re-opens in the same state
- All strings are converted into appropriate languages whenever a language translation facility is available
- Application items are always synchronized according to the user actions
- End user is provided with a user manual which helps the end user to understand and operate the application who may be not familiar with the application's proceedings

Compatibility Testing

- Compatibility testing on mobile devices is performed to ensure that since mobile devices have different size, resolution, screen, version and hardware so the application should be tested across all the devices to ensure that the application works as desired

Compatibility Testing Scenarios

- User Interface of the application is as per the screen size of the device, no text/control is partially invisible or inaccessible
- Text is readable for all users for the application
- Call/alarm functionality is enabled whenever the application is running. The application is minimized or suspended on the event of a call and then whenever the call stops the application is resumed

Recoverability Testing Scenarios

- Crash recovery and transaction interruptions
- Validation of the effective application recovery situation post unexpected interruption/crash scenarios
- Verification of how the application handles a transaction during a power failure (i.e. Battery dies or a sudden manual shutdown of the device)
- Validation of the process where the connection is suspended, the system needs to re-establish for recovering the data directly affected by the suspended connection

Other Testing Scenarios (1)

- Installation testing
  - whether the application can be installed in a reasonable amount of time and with required criterion
- Uninstallation testing
  - whether the application can be uninstalled in a reasonable amount of time and with required criterion
- Network test cases
  - validation of whether the network is performing under required load or not, whether the network is able to support all the necessary applications during the testing procedures

Other Testing Scenarios (2)

- Continued keypad entry during interrupts and other times like network issues
- Methods which deal with exiting the application
- Charger effect while an application is running in the background
- Low battery and high performance demand
- Removal of battery while an application is being performed
- Consumption of battery by application
- Check Application side effects

Manual or Automated Mobile App Testing Tools

- Both manual and automated testing are widespread and can provide good results.
- The choice should be based on the company's preferences and capabilities.
- Somebody may think that manual testing will be entirely replaced by test automation, but it is not true.
- Moreover, there are certain situations when manual control is a better option.

Manual vs Automated Mobile App Testing

	MANUAL TESTING	AUTOMATED TESTING
ADVANTAGES	• Cost-effective in a short-term period • More flexible • Better simulation of user actions	• Cost-effective in the long-term • The running process is faster • Can be reused easily • Test results are easy to share
DISADVANTAGES	• Difficult to reuse • The running process is slow • Some cases can't be executed manually	• Automated tools may have limitations • Less efficient in the defining user-friendliness or positive customer experience

Universal Mobile Testing Tools

- There are universal testing tools suitable for Android and iOS solutions.
- Some examples are:
  - Eggplant — an industry-leading test automation and monitoring platform.
  - Ranorex — an automated tool for mobile, web and desktop solutions.
  - Appium — an open-source test automation framework for mobile native, hybrid and web solutions.
  - Kobiton — an AI-powered tool for manual and automated testing.
- There are also tools specifically used for Android- only or iOS-only apps.
  - Selendroid — a test automation framework for native or hybrid Android apps.
  - iOS Driver — an open-source tool for iOS native, hybrid, or mobile web solutions.

Summary

- What is a Mobile Application?
- Mobile Testing vs. Mobile Application Testing
- Why Perform Mobile Application Testing
- Challenges in Mobile Application Testing
- Types of Mobile Application Testing