

Cross Browser Testing & Mobile Testing

INFO6255 Software Quality Control & Management

Medi Servattalab

M.Servattalab@northeastern.edu

Cross Browser Testing

- **Cross Browser Testing** is a process to test the Web applications across multiple browsers.
- **Cross browser testing** involves checking compatibility of your application across multiple web browsers and that your web application works correctly across different web browsers.

Why Cross Browser Testing is Important?



- With such a wide range of **browsers**, devices, and operating systems available today, **cross browser testing** is a crucial part of developing software.
- The purpose of **cross-browser testing** is to provide a consistent behavior and experience across all **browsers**, devices, and platforms.

Why Testing Different Browsers?



Browsers may have bugs



Difference in feature implementations



Different levels of new feature implementation



Devices constraints like hardware capabilities and display size

Month-by-Month Browser Statistics

2019	Chrome	Edge/IE	Firefox	Safari	Opera
May	80.4 %	3.6 %	9.5 %	3.3 %	1.7 %
April	80.3 %	3.6 %	9.6 %	3.3 %	1.7 %
March	80.0 %	3.8 %	9.6 %	3.3 %	1.7 %
February	79.7 %	4.0 %	10.0 %	3.4 %	1.6 %
January	79.5 %	4.0 %	10.2 %	3.3 %	1.6 %

Cross Browser Testing - What to test?

1. CSS validation (Cascading Style Sheet)
2. HTML or XHTML validation
3. Page validations with and without JavaScript enabled
4. Ajax and JQuery functionality
5. Font size validation
6. Page layout in different resolutions



- **Repeat the testing:**

- Different **Operating Systems** like Windows, Linux, and Mac
- Different **browsers** (with different versions) like Internet Explorer, Firefox, Google Chrome, Safari, and Opera.

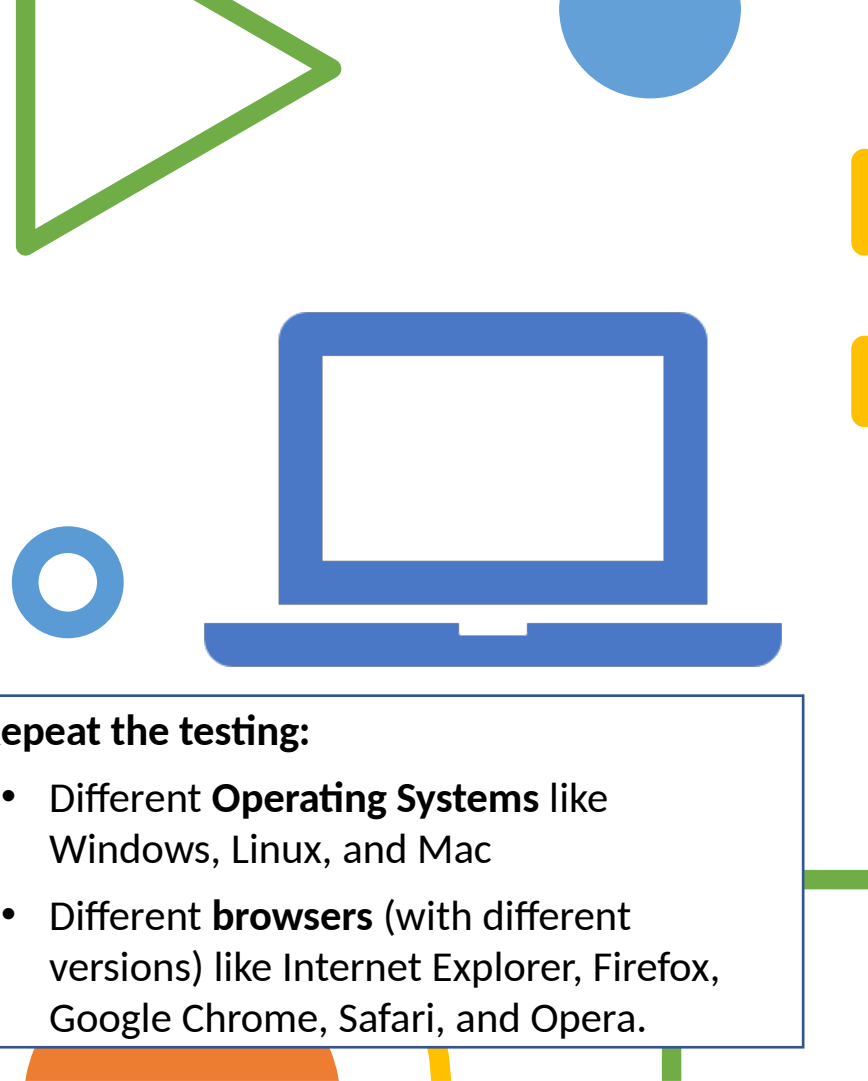
Cross Browser Testing

- What to test?

7. All images and alignment
8. Header and footer sections
9. Page content alignment to center, LHS or RHS
10. Page styles
11. Date formats
12. Special characters with HTML character encoding
13. Page zoom-in and zoom-out functionality

• Repeat the testing:

- Different **Operating Systems** like Windows, Linux, and Mac
- Different **browsers** (with different versions) like Internet Explorer, Firefox, Google Chrome, Safari, and Opera.



Cross Browser Testing Tools

Saucelabs

Browserstack Live

BrowserShots

EndTest

Selenium Grid

[Home](#)[Queue](#)[Factories](#)[Recent](#)[Browser Compatibility Test](#)[Web Design Gallery](#)[Icon S](#)

Enter URL Here:



Linux



Windows



Mac

☐ Arora 0.1☒ Arora 0.11☒ Chrome 37.0☒ Chrome 38.0☒ Chrome 48.0☒ Chrome 69.0☒ Firefox 38.0☒ Firefox 39.0☒ Firefox 40.0☒ Firefox 43.0☒ Firefox 44.0☒ Firefox 50.0☒ Chrome 39.0☒ Chrome 44.0☒ Chrome 45.0☒ Chrome 51.0☒ Chrome 71.0☒ Firefox 30.0☒ Firefox 41.0☒ Firefox 42.0☒ Firefox 43.0☒ Firefox 44.0☒ Firefox 45.0☒ Firefox 46.0☒ Firefox 49.0☒ Firefox 50.0☒ Firefox 60.0☒ Firefox 61.0☒ Firefox 62.0☒ Firefox 63.0☒ Chrome 41.0☒ Chrome 45.0☒ Chrome 48.0☒ Firefox 30.0☒ Firefox 31.0☒ Firefox 32.0



More on BrowserShots

- Run **cross-browser compatibility tests** with great customization options like
 - Browser type
 - Operating system
 - Screen size
 - Color depth
 - JavaScript status
 - Flash enable/disable settings
- Enter: website URL, compatibility test parameters and Submit the test request.
- This free browser compatibility test service can be used for taking website screen-shots almost in **61 browsers and various operating systems**.



References

https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Cross_browser_testing

<https://www.w3.org/WAI/intro/people-use-web/browsing>

<https://www.w3schools.com/browsers/>

<https://notes.mobiterra.com/react-native-part-1-the-problems-1ba30ff49ad7#.4svvvukmy>

https://www.tutorialspoint.com/mobile_testing/mobile_testing_tutorial.pdf

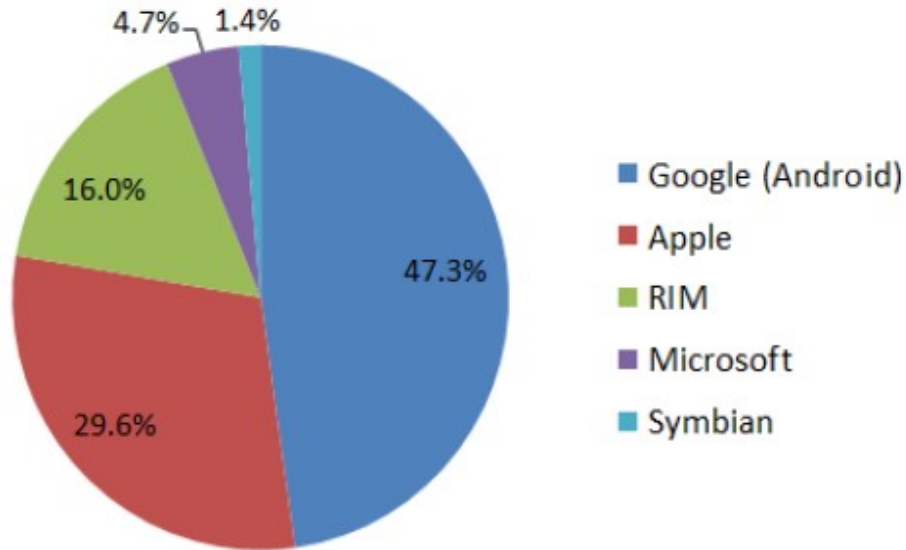
Mobile Testing

Mobile Operating Systems

The following table gives an overview of some of the popular mobile operating systems available in market:

Operating System	Developed by	Popularity (Low, Medium, High)	Latest available version
Android	Google Inc	High	Lollipop, Android 5.0-5.1
iOS	Apple Inc	High	iOS 8.X
Blackberry	Blackberry Ltd	Low	Blackberry 10.2.1
Windows	Microsoft Inc	Medium	Windows 10 Mobile
Symbian	Symbian Foundation	Low	Discontinued

Based on some generic survey, the usage of different operating systems in the market be depicted as shown below.



Mobile
Popularity

Mobile Device Ty

Device	Tablets	E-book Readers	Smartphones
What it is	Tablets are portable computer devices. Unlike traditional computers, they don't have keyboards or mouse, however the entire screen is touch sensitive.	E-book readers—also called e-readers—are similar to tablet computers, <u>except they are mainly designed for reading e-books (digital, downloadable books).</u>	A smartphone is a powerful mobile phone that is designed to run a variety of applications in addition to providing phone service.
Used for	Almost all the jobs which we can do with traditional computers or desktops.	Reading e-books	Web browsing, watching videos, reading e-books, and playing games
Example	Samsung Tablets	Amazon Kindle, Barnes & Noble Nook.	Sony smartphones, Samsung smartphones, Apple iPhone.

What are the different Application Types?

- The Different Application Types:
 - Native Application
 - Hybrid Application
 - MobileWeb Application



Native Application

- **Native App** – Lives on the device.
- **Pros:**
 - Takes advantage of the device features (use the camera, GPS, etc.)
 - Uses the device's notification system
 - Publishes push notifications
 - Maintains the same GUI design and offer best experience
- **Cons:**
 - High cost of building the app
 - Higher maintenance

Native vs Hybrid vs MobileWeb

- **Hybrid App** – is a way to expose content from existing in App format.

- **Pros**

- Cheaper to develop
- Easy Maintenance
- Takes advantage of the new features available
- Embedded browser
- Graphics are not accustomed to the O/S

- **Cons**

- Slower performance

Native vs Hybrid vs MobileWeb

- **MobileWeb** – are not real applications. They open up in the Mobile Browsers.

- **Pros**

- Easy to develop
- Easy Update
- No Installation required
- No Icon on the mobile
- Does not take advantage of the Mobile features

- **Cons**

- Not on AppStore

Device Testing vs Application Testing

Device Testing

- Unit Testing
- Factory Testing
 - Mobile application testing
 - Hardware testing
 - Battery (charging) testing
 - Signal receiving
 - Network testing
 - Protocol testing
 - Mobile games testing
 - Mobile software compatibility testing
- Certification Testing

Application Testing

- Functional testing
- Laboratory testing
- Performance testing
- Memory Leakage testing
- Interrupt testing
- Usability testing
- Installation testing
- Certification testing
- Security testing

Real Device vs Emulator/Simulator

	Real Device	Emulator / Simulator
Price	Getting real devices will cost you a lot.	It is almost free, we just need to download and install them
Processing Speed	It has faster processing; however network latency may be normal.	It is slower as compared to actual devices. It has observed less latency than real devices connected to the local network or in the cloud.
Debugging	Debugging is not that easy.	It provides step-by-step debugging of an application. Also, it provides an efficient way for capturing screenshots.
Web-app Testing	Web applications can be tested in a normal way.	Testing a web application is much easier.
Reliability	Testing on a real device has a major advantage that it always gives accurate results.	It cannot simulate all types of user interactions; hence it may lead to false results sometimes. So it scores low when it comes to reliability.

Emulator vs Simulator

	Emulator	Simulator
What it mimics	<ul style="list-style-type: none">- Mobile device software- Mobile device hardware- Mobile operating system	<ul style="list-style-type: none">- Internal behavior of the device.- It does not mimic hardware.
How to get it	It is generally provided by the device manufacturer.	It is generally provided by the device manufacturer or some other company.
Internal structure	It is written in machine-level assembly language.	It is written in high-level language.
Debugging	It is more suitable for debugging.	It is not suitable for debugging purpose.
Performance	Emulators are really slow. Emulating the actual hardware usually makes the software run slower than it would natively.	Faster than emulators.
Example	Google's Android SDK	Apple's iOS Simulator

iOS and Android Testing Frameworks

- The Testing Framework is a code written by QA and it is processed in a special testing environment.
- iOS automation testing frameworks allow for:
 - Defining the format in which to express expectations.
 - Creating a mechanism to hook into or drive the AUT Executing the tests.
 - Reporting results.

Android Testing Frameworks

- **Robotium:** is an open-source test framework for developing functional, system and acceptance test scenarios. **It is very similar to Selenium.**
- **UIAutomator:** is a test framework by Google that provides advance UI testing of native Android apps and games. It has a Java library containing API to create functional UI tests and also an execution engine to run the tests.
- **Appium:** is an open-source test automation framework to test native and hybrid apps and mobile web apps.
 - Appium library functions inside the framework make calls to the Appium server running in the background which operates the connected device.

Android Testing Frameworks (Continued)

- **Calabash:** is a functional testing framework that can be used for both iOS and Android functional testing.
 - On paper, it must be one of the easiest frameworks to use and even non-developers should be able to create functional tests using it.
- **Selendroid:** is a relatively new kid on the block and can be used to functionally test your Android applications.
 - Apparently, if you are used to Selenium, Selendroid should be an easy way to use your knowledge to create your functional tests for Android.

iOS Testing Frameworks

Appium: is an open-source test automation framework to test native and hybrid apps and mobile web apps.

Appium library functions inside the framework make calls to the Appium server running in background which operates the connected device.

Calabash: is a functional testing framework that can be used for both iOS and Android functional testing.

On paper, it must be one of the easiest frameworks to use and even non-developers should be able to create functional tests using it.

iOS Testing Frameworks

Zucchini: is an open-source visual functional testing framework for iOS applications based on Apple UIAutomation.

UI Automation: For your more typical functional tests (or black-box tests), in which you're going to write code that simulates an end-user navigating your app, there is UI Automation.

UI Automation is provided by Apple and is the Apple-sanctioned way of performing iOS functional testing.

FRANK - for iOS: If you want to do end-to-end testing in iOS and wish you could use Cucumber, no worries — there's a tool called **Frank** that will allow you to create acceptance tests and requirements using Cucumber

Testing Frameworks

Comparison between different testing frameworks

	Robotium	uiautomator	Espresso	Appium	Calabash
Android	Yes	Yes	Yes	Yes	Yes
iOS	No	No	No	Yes	Yes
Mobile web	Yes (Android)	Limited to x.y clicks	No	Yes (Android & iOS)	Yes (Android)
Scripting Language	Java	Java	Java	Almost any	Ruby
Test creation tools	Testdroid Recorder	UI Automator viewer	Hierarchy Viewer	Appium.app	CLI
Supported API levels	All	16 =>	8, 10, 15 =>	All	All
Community	Contributors	Google	Google	Active	Pretty quiet

SauceLabs



Sauce Labs provides cloud-based testing for mobile and web applications. It allows running tests for more than 700 different browsers, OS and device combinations.



If all your tests are run on emulators, your app will fail in real-world scenarios.

Real Devices allow you to test more, like GPS, Carrier, etc.



Too many devices?

The best approach is to identify the most-used devices among your user base and test on them.



Maintaining a device lab in-house is expensive, inefficient, and distracts you from the tests you need to run. Instead, renting real mobile devices in the cloud by the minute is a better alternative.



Automate the testing. Performance and load testing is still required.

Mobile Testing references

- https://www.tutorialspoint.com/mobile_testing/mobile_testing_tutorial.pdf
- <https://www.crunchbase.com/organization/sauce-labs>