Testing types

1. White box testing
2. Black box testing
3. Automation (Grey box testing)

BLACK BOX TESTING

- FUNCTIONAL

Positive

- Ad Hoc
- Smoke
- Acceptance
- Regression
- Boundary Condition
- Alpha/Beta

NEGATIVE

- Error Handling

- Non Functional

- Performance
- Load
- Stress
- Security
- Compatibility
- System
- Usability
- Recovery
- Installation

White box testing

- Unit testing – test individual modules
- API testing – write calls to the backend
- Penetration testing (hacking)
- Integration testing (module + module)

AUTOMATION TESTING

Tools

- Selenium
- Calabash
- Appium

EDI

- Eclipse
- RubyMiner
- Cucumber

Functional testing

- **Ad Hoc** – testing performed without planning and documentation
- **Smoke** – A smoke test is basically just a sanity check to see if the software functions on the most basic level
- **Acceptance test** – acceptance testing is a test conducted to determine if the requirements of a specification are met (that is, the main flows of the application are functional)
- **Regression test –** seeks to uncover new software bugs, or regressions, in existing functionality of a system after changes such as enhancements or patches have been made. Also includes retesting/regression of the fixed issues (bugs)
- Boundary testing – or, boundary value analysis, is where test cases are generated using the extremes of the input domain, e.g. maximum, minimum
- **Alpha** - testing is simulated, or actual operational testing by potential users/customers, or an independent test team at the developers' site.

Beta - testing comes after alpha testing and can be considered a form of external user acceptance testing.

- **Error Handling (negative) –** test designed to determine the response of the system outside of what is defined.

Functional testing

- **Ad Hoc** – testing performed without planning and documentation
- **Smoke** – A smoke test is basically just a sanity check to see if the software functions on the most basic level
- **Acceptance test** – acceptance testing is a test conducted to determine if the requirements of a specification are met (that is, the main flows of the application are functional)
- **Regression test –** seeks to uncover new software bugs, or regressions, in existing functionality of a system after changes such as enhancements or patches have been made. Also includes retesting/regression of the fixed issues (bugs)
- Boundary testing – or, boundary value analysis, is where test cases are generated using the extremes of the input domain, e.g. maximum, minimum
- **Alpha** - testing is simulated, or actual operational testing by potential users/customers, or an independent test team at the developers' site.

Beta - testing comes after alpha testing and can be considered a form of external user acceptance testing.

- **Error Handling (negative) –** test designed to determine the response of the system outside of what is defined.

Functional testing

- **Ad Hoc** – testing performed without planning and documentation
- **Smoke** - A smoke test is basically just a sanity check to see if the software functions on the most basic level
- **Acceptance test** - acceptance testing is a test conducted to determine if the requirements of a specification are met (Main flows of the Application are functional)
- **Regression test** - seeks to uncover new software bugs, or regressions, in existing functional a system after changes such as enhancements or patches have been made to them. Also including retesting/regression of the fixed issues (bugs)
- **Boundary testing or boundary value analysis, is where test cases are generated using the extremes of the input domain, e.g. maximum, minimum**
- **Alpha** - testing is simulated or actual operational testing by potential users/customers or an independent test team at the developers' site.

Beta - testing comes after alpha testing and can be considered a form of external user acceptance testing.

- **Error Handling(negative)-** test designed to determine the response of the system outside of what is defined.

Non Functional

- **Security** - is a process to determine that an information system protects data and maintains functionality as intended.

The six basic security concepts that need to be covered by security testing are:

- **Confidentiality** - A security measure which protects against the disclosure of information to parties other than the intended recipient that is by no means the only way of ensuring the security
- **Integrity** -A measure intended to allow the receiver to determine that the information provided by a system is correct
- **Authentication**-This might involve confirming the identity of a person, ensuring that a product is what its labeling claims to be, or assuring that a computer program is a trusted one.
- **Availability**-Assuring information and communications services will be ready for use when expected.
- **Authorization** - The process of determining that a requester is allowed to receive a service or perform an operation
- **Non-repudiation** - In reference to digital security, nonrepudiation means to ensure that a transferred message has been sent and received by the parties claiming to have sent and received the message.

Non Functional

- **Load Test** - is the process of putting demand on a system or device and measuring its response
- **Performance test** - is in general testing performed to determine how a system performs in terms of responsiveness and stability under a particular workload
- **Stress test**- is a form of deliberately intense or thorough testing used to determine the stability of a given system or entity
- **Compatibility**- is testing conducted on the application to evaluate the application's compatibility with the computing environment.
- **Recovery testing** is the activity of testing how well an application is able to recover from crashes, hardware failures and other similar problems.
- Usability - Usability testing focuses on measuring a human-made product's capacity to meet its intended purpose
- **System -** testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements
- **Installation - focuses on what customers will need to do to install and set up the new software successfully. The testing process may involve full, partial or upgrades install/uninstall processes**

Mobile testing focus

- Like any desktop or web application testing, mobile application testing is also focused on the quality and performance of the final product. However, mobile app testing becomes far more challenging because of the following key factors:

Mobile testing challenges

Portability----
Mobile application testing is diff cult due to compatibility issues as a mobile application can be deployed across devices which have different:

- **Operating systems**
like iOS,Android,Windows,BB, JAVA etc.

- **Versions of operating system**
such as iOS 4.x,iOS 5.x Android 4.0, 4,1 4.2 and so on

- **Manufacturers** like Samsung,HTC,Nokia,Micromax,etc.

- **Keypad type** such as virtual keypad, hardkeypad, etc.

Mobile Network Operator-----CDMA & GSM

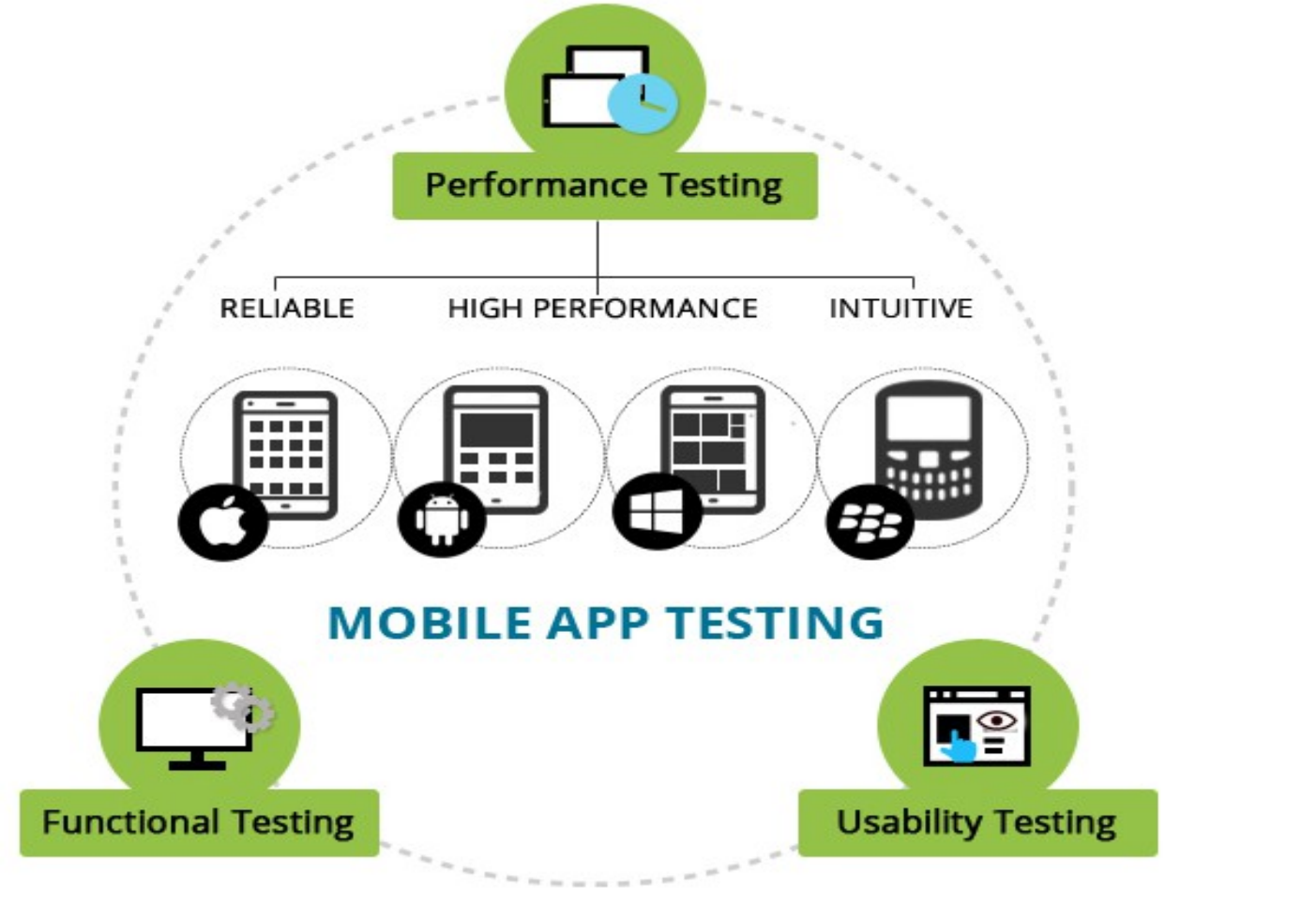
Further, the quality team cannot guarantee that if a tested application works well on a given device, it will work 100% on another device even if it is from the same product family because the screen resolution, CPU, Memory, OS optimization, and hardware could be different.

-

Mobile Testing Challenges 2

- Flakey connectivity
- Log in on multi devices
- Multiple networks (Wi-Fi, 4G, 3G, etc.)
- Battery life
- Web View testing on mobiles

Mobile testing strategy



Compatibility testing example

AVA

- Install / Uninstall – different devices/ ios
- New Install vs Old Install – Android and iPhone
- Old Install vs Update – Android and iPhone
- New Install vs Update– Android and iPhone
- Update vs Update – Android and iPhone
- New Install vs New Install – Android and iPhone

Main Flows Examples

- Send Video using txt Android → iPhone → Android
- Send Video using WhatsApp Android → iPhone → Android
- Send Video using Skype Android → iPhone → Android
- Send Video using Kik Android → iPhone → Android
- Receive video on iPhone that was sent using different messengers
- Receive video on Android that was sent using different messengers
- Play video (Android, iPhone, Tablets)
- Record video (Android, iPhone, Tablets)
- Share video using different messengers to iPhone
- Share video using different messengers to Android
- Send video using different messengers from iPhone → iPhone (no App Installed)
- Send video using different messengers from iPhone → Android (no App Installed)

Main Flow Examples

AVA

- Send video using different messengers from Android → iPhone (no App Installed)
- Send Video to multi users
- Save video
- Delete vide
- Send Invite
- Test with Wifi On/Off
- Send Long Videos
- Send videos with Notifications On/Off

Test Case Template

Purpose	Video can be sent from iPhone to iPhone using txt	Automation	Screen Shots
Steps to reproduce	1. Launch the App 2. Tap and start recording video 3. Tap to stop recording 4. Click the Next button 5. Select Contact 6. Click the “Send” button	Yes	
Expected results:	Video has been sent message Video is received		
Notes	This test case should pass on different iPhone/ios configurations		