Becoming a Pro

IN Mobile Applications Testing

Mobile APPS: Distribution/Installation: Android .APK

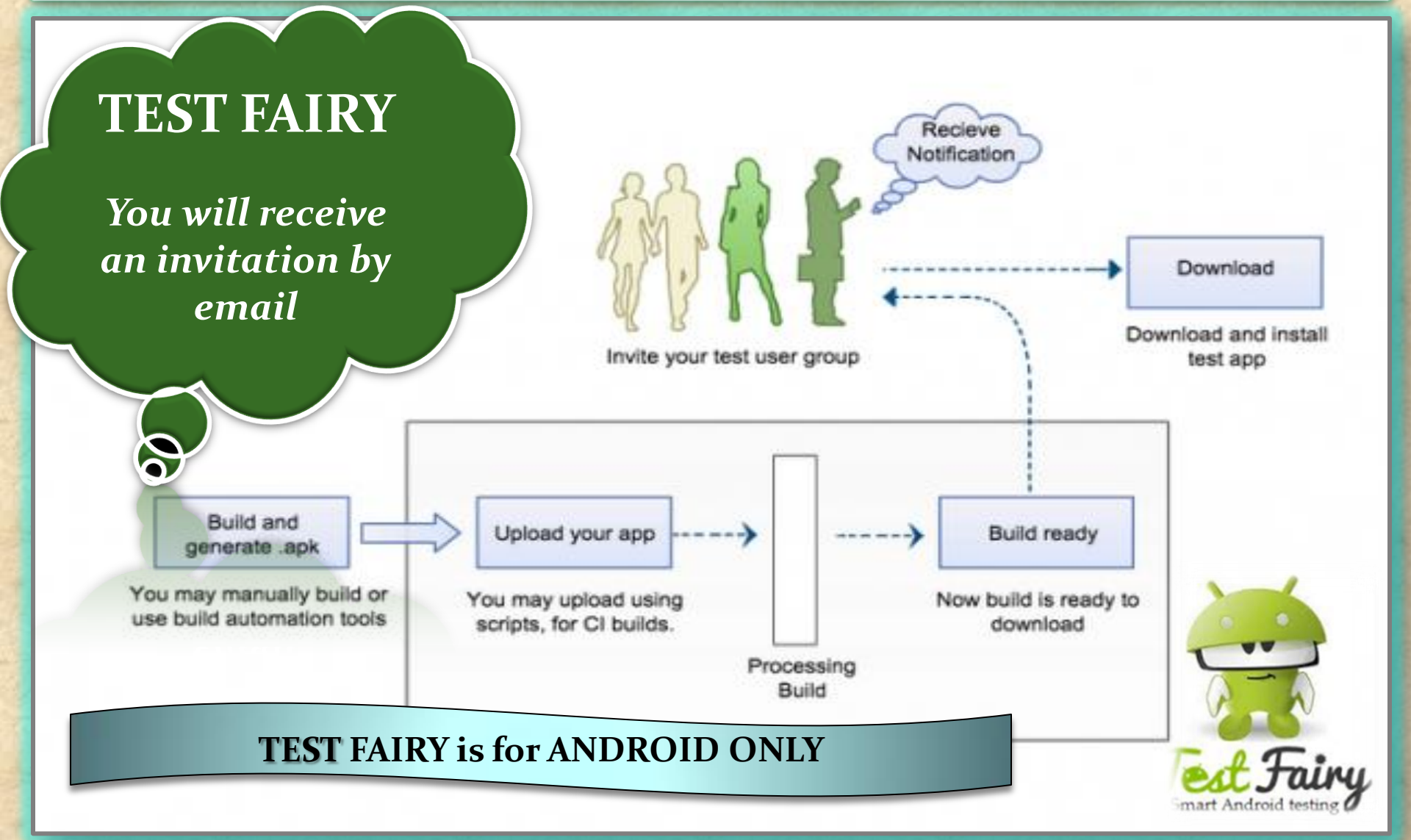
TestFairy offers some great features for app developers. One of the stand out features is client side Video recording and not just screen shots.

TestFairy provides a video recording of the exact test from the client side, including CPU, Memory, GPS, Network monitoring, logs, crash reports and more.

With Android apps you don't need to integrate any SDK or APIs into your app builds. You upload your APK (Android application file) to the TestFairy platform.

TEST FAIRY is for ANDROID ONLY

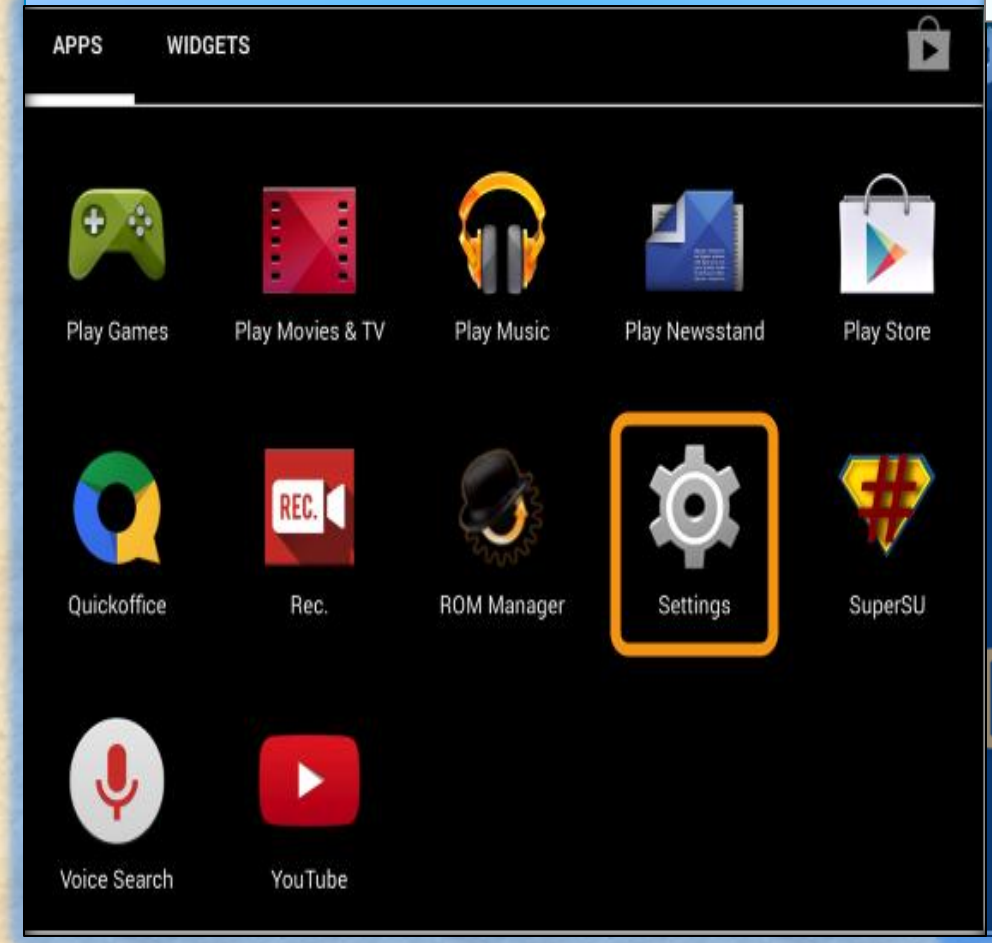
Mobile APPS: Distribution/Installation: Android .APK



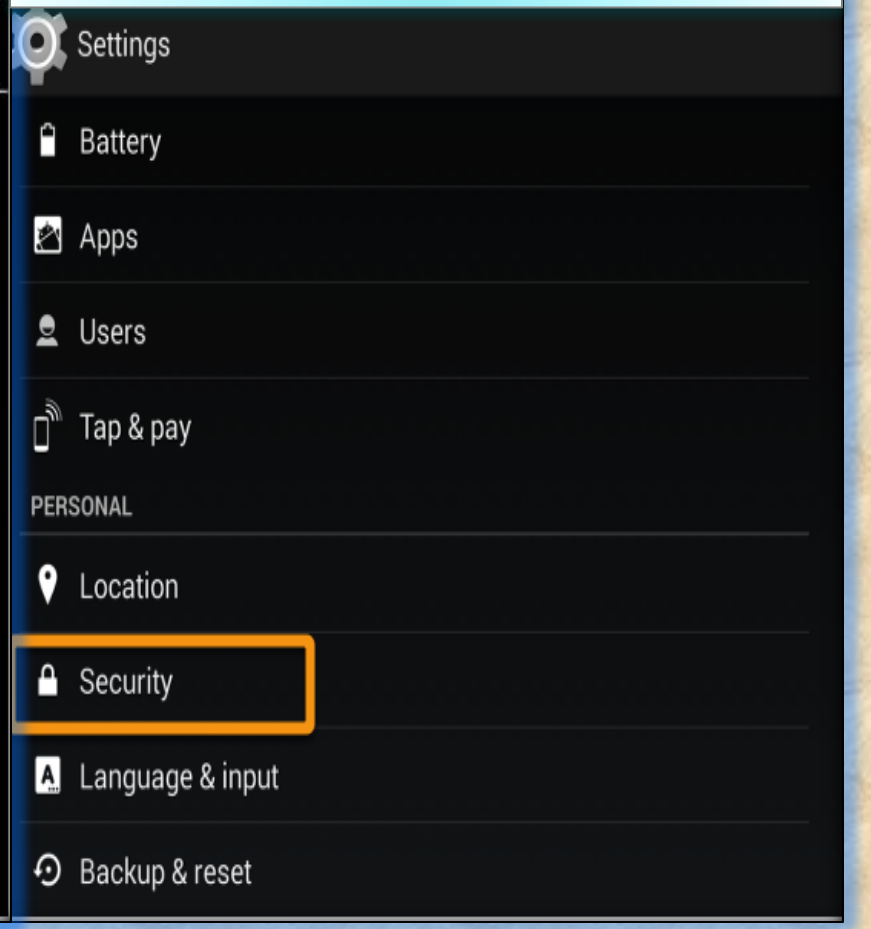
Mobile APPS: Distribution/Installation: Android .APK

Installing THIRD PARTY APK

1. Go to the "Settings" application on your Android device



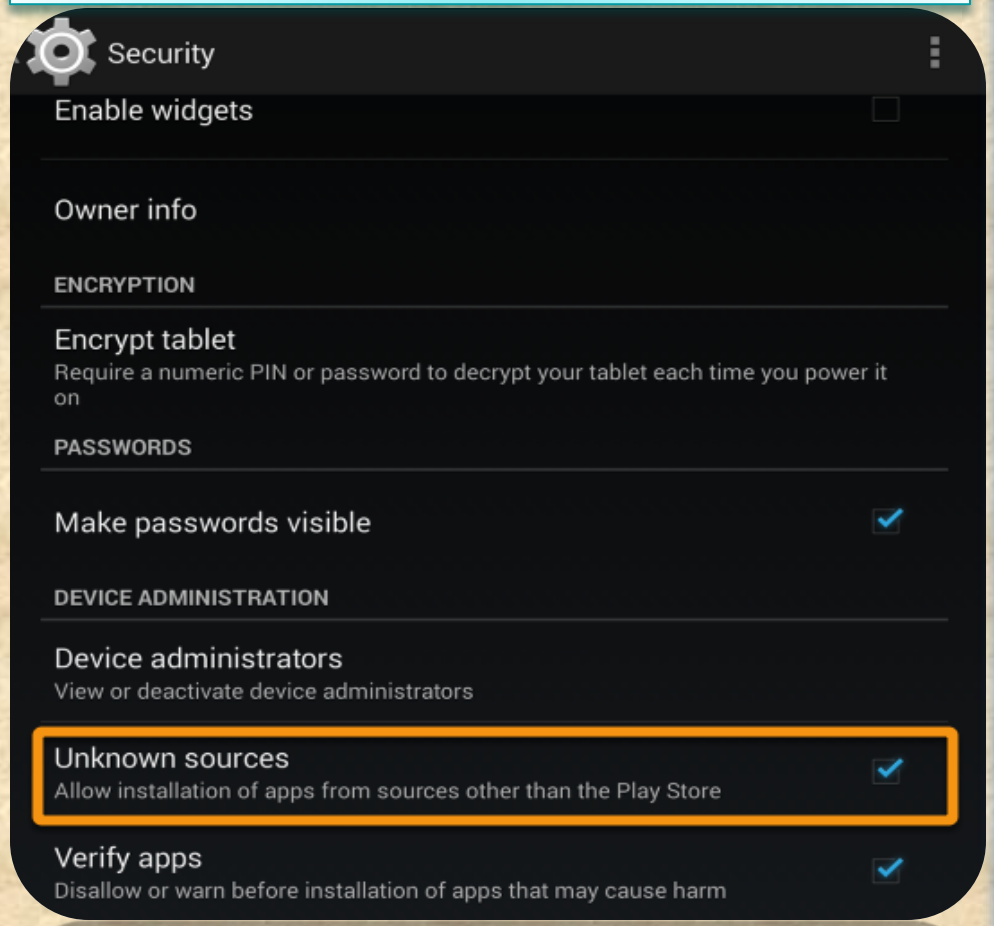
2. Choose the "Security" option located under the heading "Personal"



Mobile APPS: Distribution/Installation: Android .APK

Installing THIRD PARTY APK

3. Under "Device Administration" place a checkbox next to the option "Unknown Sources"



This allows you to install applications on your Android device that are not downloaded directly from the Google Play store

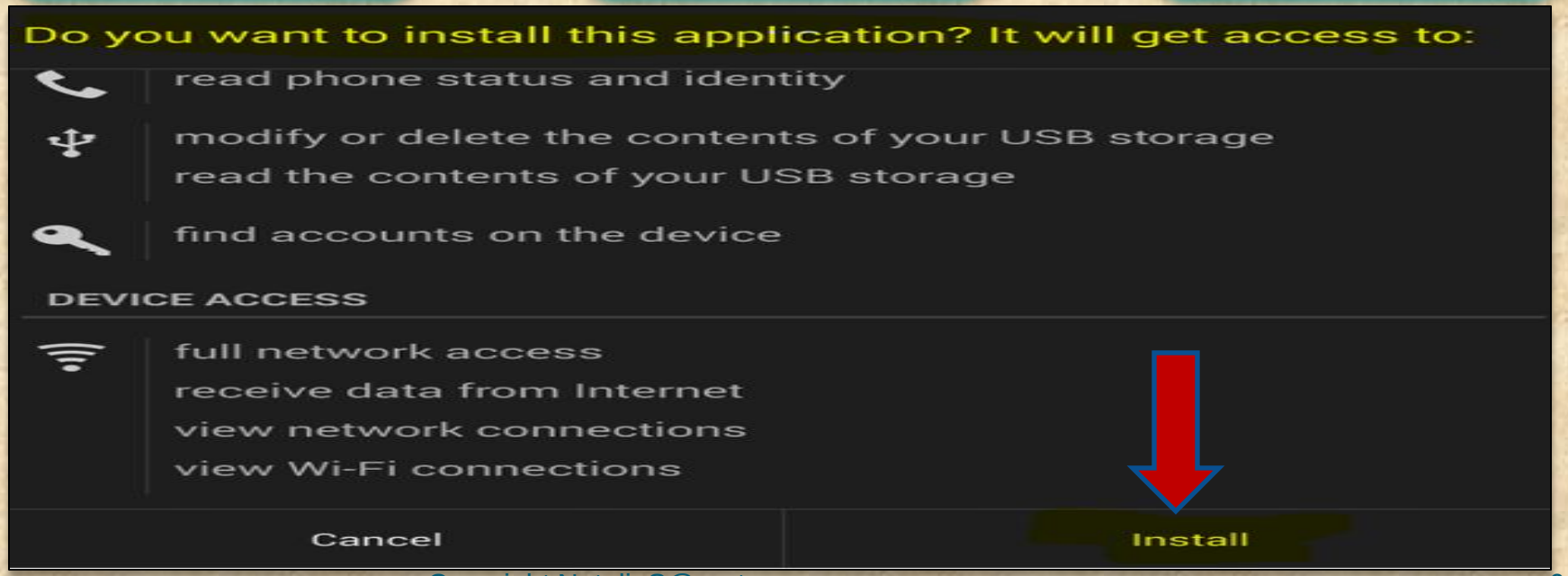
Mobile APPS: Distribution/Installation: Android .APK

Transferring APK to your Device

1. On your computer, attach the ".apk" file to an e-mail and send it to an account that you can access via your Android device.

2. On the Android device, click on the ".apk" attachment in the e-mail in order to download it.

3. Follow the on-screen instructions to install the application.



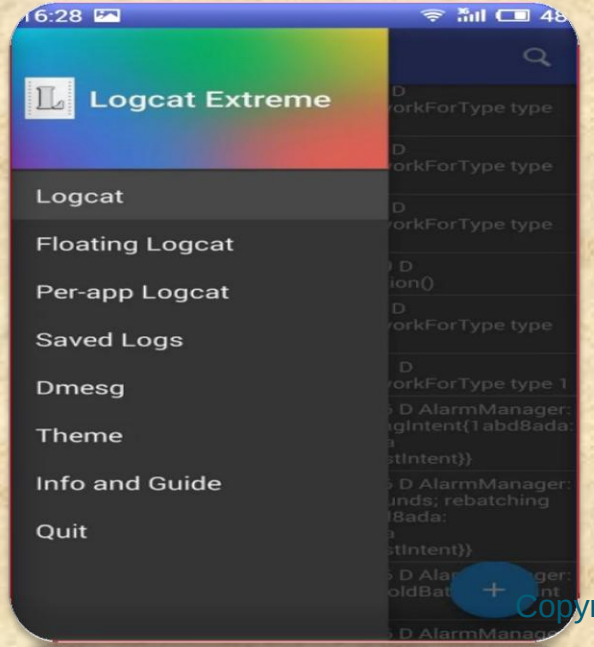
Mobile APPS: Android LOGS : EXTRA

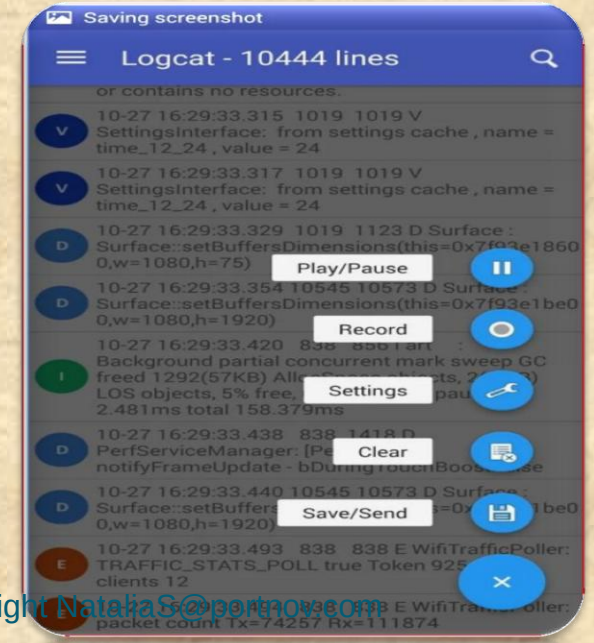
Installing LOGCAT APP to your Device

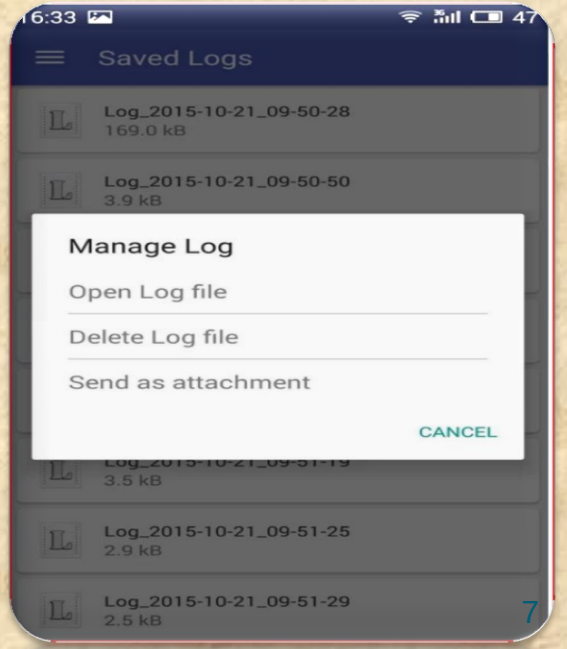
Download **LOGCAT EXTREME** from PlayStore to your Device directly *(this is only an example, there are many other similar apps)*

LOGCAT EXTREME *is* an enhanced Logcat reader and Logcat recorder which comes with a rich set of features and handy user interface.

Please note: From Android 4.1 onwards ANY logcat app needs root access in order to show logs properly.







Mobile APPS: Distribution/Installation: iOS .IPA

Through an App Marketplace (APP STORE)

XCODE

ITUNES

TEST FLIGHT

Mobile APPS: Distribution/Installation: .IPA

What is XCode?

Xcode is an Intergrated Development Environment by Apple containing a suite of software Development Tools for macOS, iOS, WatchOS and tvOS

Xcode supports source code for programing languages C, C++, Objective-C, Objective-C++, Java, AppleScript, Python, Ruby, ResEdit (Rez), and Swift

Also supports variety of programming models, including but not limited to Cocoa, Carbon, and Java

Mobile APPS: Distribution/Installation: .IPA

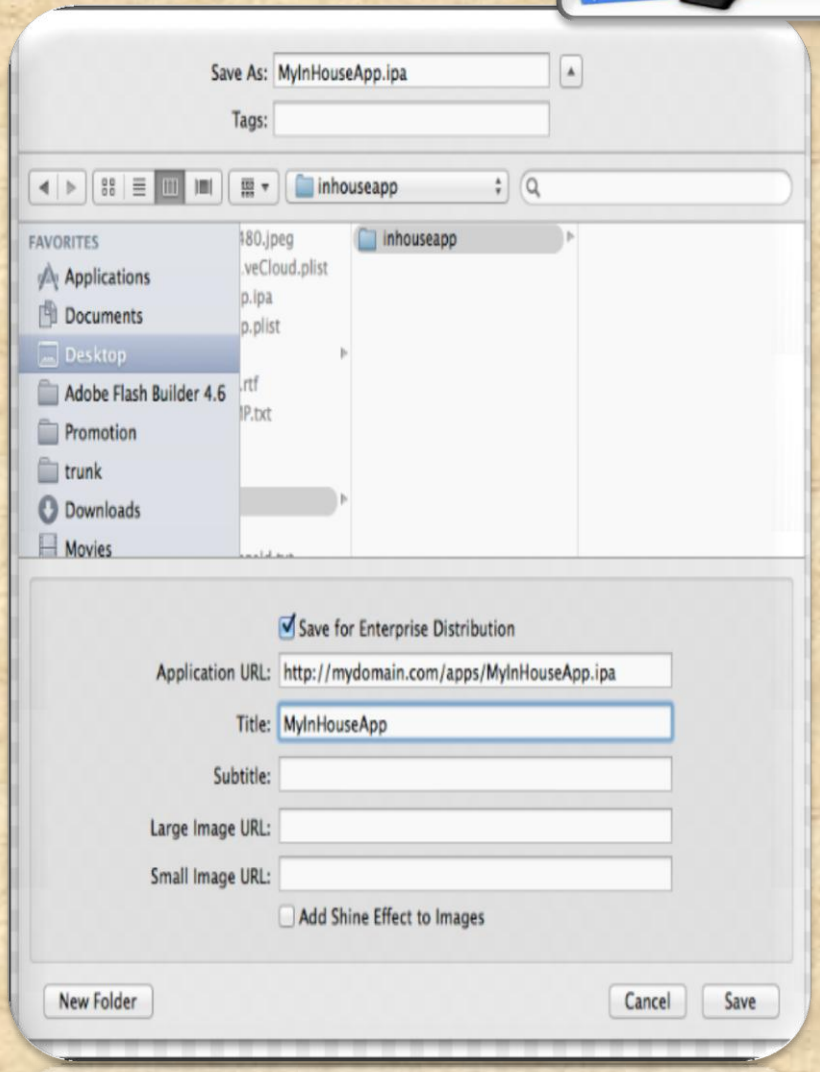
Distributing .IPA through XCode ?

Connect the device to your Mac.

In Xcode, choose Window > Devices and select the device under Devices.

In the Installed Apps table, click the Add button (+) below the table.

In the dialog that appears, choose the iOS App file and click Open.



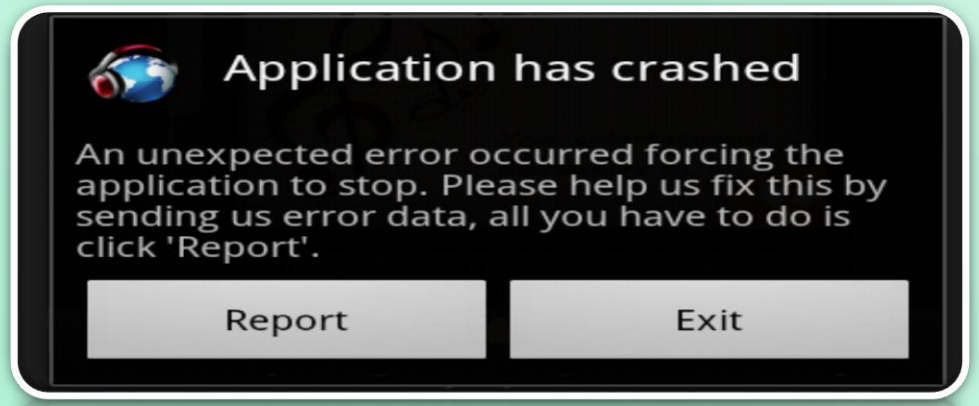
Mobile APPS: Collecting LOGS: .IPA

How to do it through Xcode on MAC ?

1. Install XCode
2. Connect your iPhone to the Mac
3. Select Trust this computer on the iPhone pop-up request
4. Start xCode (Menu) →Window → Devices (Select your iPhone and press the arrow button in the bottom right
5. Reproduce the problem
6. Press the arrow button and download the logs

How can I debug a deployed app without Xcode debugger ?

- A: Once you have deployed your app, either through the *App Store* or as an *Ad Hoc* or *Enterprise* build, you won't be able to attach *Xcode's* debugger to it.
- To debug problems, you need to analyze *Crash Logs and Console* output from the device.



Getting Crash Logs and Console Output

Getting Crash Logs Directly From a Device

Without Xcode

Users can retrieve crash reports from their device and send them to you via email by following these instructions.

(It is not possible to get device console logs directly from a device)

- Open Settings app
- Go to Privacy, then Diagnostics & Usage
- Select Diagnostics & Usage Data
- Locate the log for the crashed app. The logs will be named in the format: *<AppName>_<DateTime>_<DeviceName>*
- Select the desired log. Then, using the text selection UI select the entire text of the log. Once the text is selected, tap Copy
- Paste the copied text to Mail and send to an email address as desired

Mobile APPS: Distribution/Installation: .IPA

What is iTunes?

iTunes is a media player, media library, online radio broadcaster, and mobile device management application developed by Apple Inc.

It is used to play, download, and organize digital downloads of music and video (as well as other types of media available on the iTunes Store) on personal computers running the macOS and Microsoft Windows operating systems.

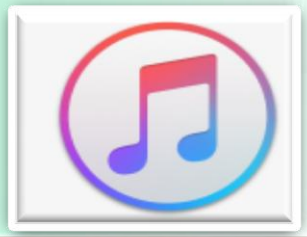
Mobile APPS: Distribution/Installation: .IPA

- *Drag-and-drop IPA* file into '*Apps*' tab of iTunes BEFORE you connect the device 
- Connect your device
- Select *your device* on iTunes
- Select '*Apps*' tab
- *Search app* that you want to install
- Click on '*Install*' button. This will change to '*Will Install*'
- Click on '*Apply*' button on right corner

LOGS: Collecting from iTunes

- Sync your device with iTunes on your desktop.
- After syncing, look for crash logs in the correct directory.

The Following few slides will give an instructions . Lets start !



LOGS: Collecting from iTunes

- **Mac OS X:**

- Open Finder (found in the Dock)
- Click on the 'Go' menu at the top of your screen, and select 'Go to Folder'
- Type (or paste): *~/Library/Logs/CrashReporter/MobileDevice/<DEVICE_NAME>*
- Open the folder with the same 'name' as your device.

  (Note: your device name appears in iTunes on the left side, under 'Devices').
- Open the folder called 'Retired'
- You will see at least one item starting with 'ReadItLaterPro'.

LOGS: Collecting from iTunes

- **Windows Vista or 7:**

- Open any Windows Explorer Window (My Computer, My Documents, etc.)
- Enter %appdata%, and press enter →Navigate to Roaming ***C:\Users\<USERNAME>\AppData\Roaming\Apple***
- ***Computer\Logs\CrashReporter\MobileDevice\<DEVICE_NAME>***
- (Note: your device name appears in iTunes on the left side, under 'Devices')
- You will see at least one item starting with 'ReadItLaterPro'.

LOGS: Syncing your device with iTunes

Windows XP:

- Locate your Application Data folder.
- Navigate to Apple computer ***C:\Documents and Settings\<USERNAME>\Application Data\Apple***
- ***Computer\Logs\CrashReporter\MobileDevice\<DEVICE_NAME>***
- (Note: your device name appears in iTunes on the left side, under 'Devices')
- You will see at least one item starting with 'ReadItLaterPro'.

LOGS: Collecting from your Device

- To obtain iOS crash logs, please select your device and go to:

- ***Settings > Privacy >Analytics>Analytics Data (for iOS 11 or newer)***
- ***Settings > Privacy > Diagnostics & Usage (for iOS 8 or newer)***
- ***Settings > General > About > Diagnostics & Usage (for iOS 7 or older)***
- ***Note: some rolled updates may affect the directory name***

- Select a ***Chrome*** crash from the list.
  This will start with ***"Chrome_"*** and contain the timestamp of the crash.
- Tap on the crash and you will see a text field with a crash log.
  Long press to *Select All* and then *Copy* the crash text.
- Paste it into something you can get off of your device (for example, an email to yourself).

Mobile APPS: Distribution/Installation: .IPA

What is TestFlight?

TestFlight is an online service for over-the-air installation and testing of mobile applications, currently owned by Apple Inc and only offered to developers within the iOSDeveloper Program

Developers signed up with the service to distribute applications to internal or external beta testers, who could subsequently send feedback about the application to developers

The TestFlight SDK additionally allowes developers to receive remote logs, crash reports and tester feedback.

HOMEWORK : READ AN ENTIRE ARTICLE ABOUT TestFLIGHT
https://www.raywenderlich.com/48750/testflight-sdk-tutorial

Overview: *Mobile APPS*

- Categories
- Types
- Distribution/Installation/Logs
- Mobile Test Industry Standards
- Remote Device Access (RDA)
- Emulators
- Simulators
- Troubleshooting Guide
- App Risk Analysis

Mobile Test Industry Standards :

Testing Strategies for Mobile Apps

Challenges

- Device fragmentation
- In-house vs. outsourced testing
- Availability of mobile testing tools
- Application Lifecycle Testing

- 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 Test Industry Standards :

Testing Strategies for Mobile Apps

- Device Model
- OS Version
- Screen Resolution
- Form Factor
- Emulators vs. Physical Devices

Mobile Test Industry Standards :

Testing Strategies for Mobile Apps

- Network density
- How the app behaves on specific devices
- How real-world users interact with the app
- Different battery states on the devices
- Multiple networks (Wi-Fi, 4G, 3G, etc.)

Mobile Test Industry Standards :

Testing Strategies for Mobile Apps

- applications can be deployed, tested, and managed
- saves businesses from setting up on-premise test environments
- capability to support complex apps
- provides real-time testing results

Mobile App Testing on Cloud

Mobile Test Industry Standards :

Testing Strategies for Mobile Apps

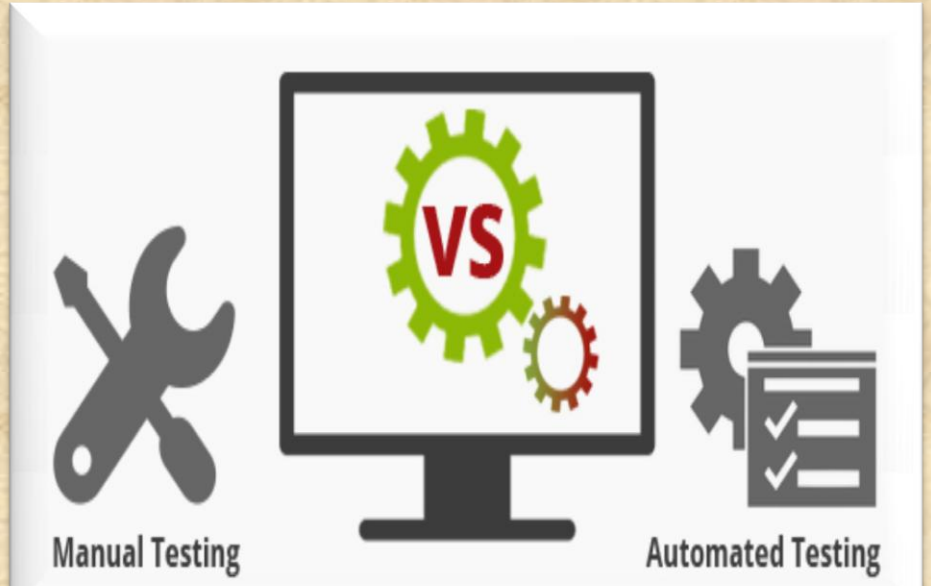
- testing mobile apps in real network environments
- network simulation tools are available
- test mobile apps in various network speeds, bandwidths variations
- testing the app in a full internet connectivity scenario and other factors

Mobile Test Industry Standards :

Testing Strategies for Mobile Apps

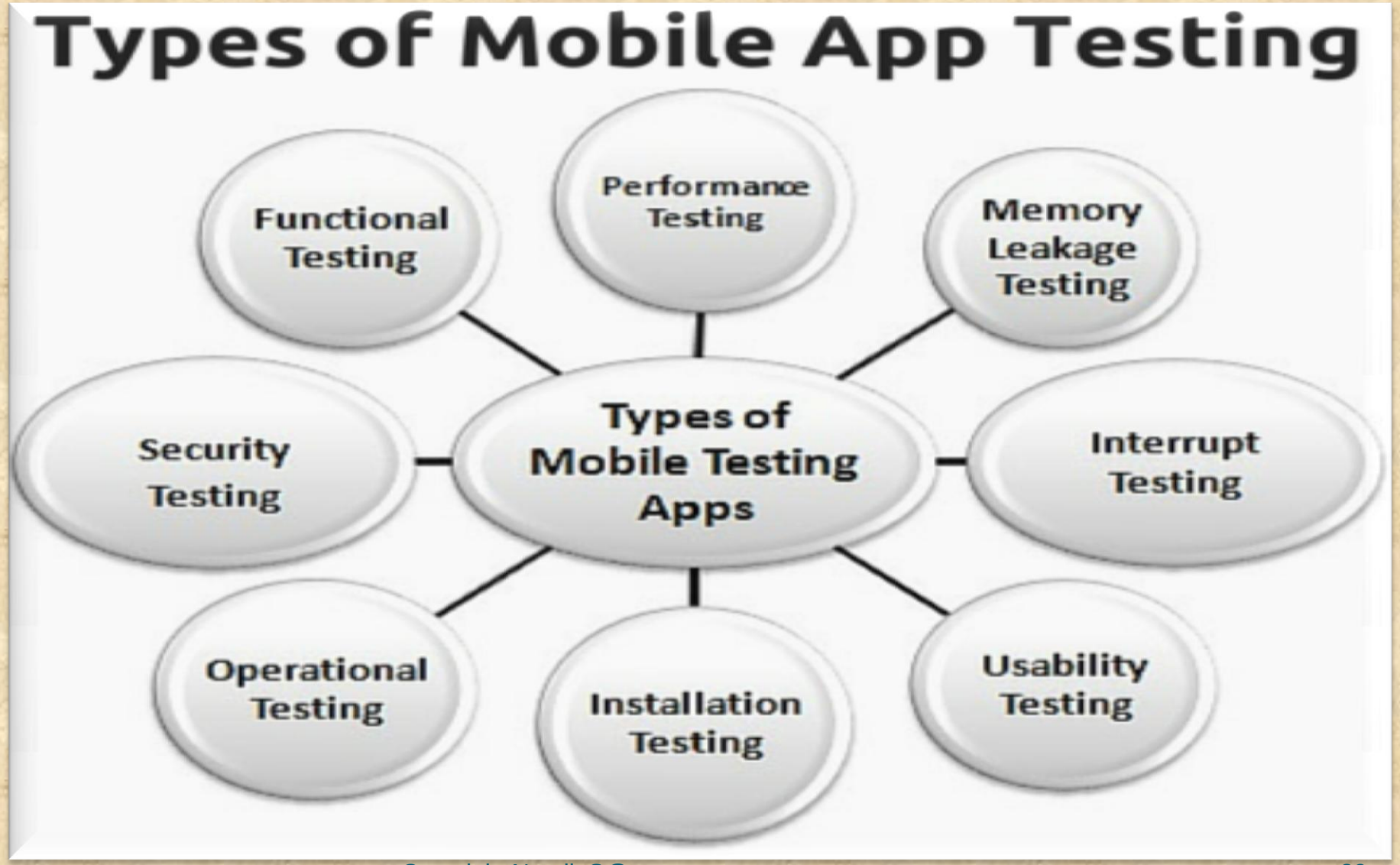
Automated testing is highly effective in consistently repeating a test procedure in regression testing as well as testing during the development stages.

However, test automation requires significant amount of initial investment.



Mobile Test Industry Standards :

Testing Strategies for Mobile Apps



Mobile Test Industry Standards :

Testing Strategies for Mobile Apps

FUNCTIONAL TEST

- Verifying that all documented requirements are implemented.
- Verifying that all features work as expected.
- Validating texts, logos, images, text captions and other UI elements.
- Validating localization and globalization.
- Evaluating ease of navigation and screen transitions.
- Examining response speed.
- Evaluating the intuitiveness of the touch interface.

Mobile Test Industry Standards :

Testing Strategies for Mobile Apps

PERFORMANCE TEST

- Performance with low battery power
- Performance while network out of coverage area
- Performance during poor bandwidth
- Performance while changing internet connection mode
- Performance while transferring heavy file
- Testing from Application's server and client side

Mobile Test Industry Standards :

Testing Strategies for Mobile Apps

Memory Leakage TEST

- Verifying if program runs for an extended time and consumes additional memory
- Verifying if memory is allocated frequently for one-time tasks
- Verifying where the program can request memory — such as shared memory that is not released
- Verifying where memory is very limited, such as in an embedded system or portable device
- Verifying where the leak occurs within the operating system or memory manager
- Verifying when a system device driver causes the leak

Mobile Test Industry Standards :

Testing Strategies for Mobile Apps

INTERRUPT TEST

- Battery low
- Battery full- when charging
- Incoming phone call
- Incoming SMS
- Incoming Alert from another mobile application
- Plugged in for charging
- Plugged out from charging
- Device shut off
- Application Update reminders
- Alarm
- Network connection loss
- Network connection restoration

Mobile Test Industry Standards :

Testing Strategies for Mobile Apps

USABILITY TEST

To ensure that the buttons should have the required size and be suitable to big fingers.

To ensure that the buttons are placed in the same section of the screen to avoid confusion to the end users.

To ensure that the icons are natural and consistent with the application.

To ensure that the buttons, which have the same function should also have the same color.

To ensure that the validation for the tapping zoom-in and zoom-out facilities should be enabled.

To ensure that the keyboard input can be minimized in an appropriate manner.

To ensure that the application provides a method for going back or undoing an action, on touching the wrong item, within an acceptable duration.

To ensure that the contextual menus are not overloaded because it has to be used quickly.