# Becoming a Pro

## IN Mobile Applications Testing

# MOBILE APPS: Lifestyle

Music apps

Travel Apps

Food & Drink apps

Dating apps

# MOBILE APPS: Social Networking

Facebook

Circles

Path

Instagram

# Overview: *Mobile APPS*

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

# MOBILE APPS: Types

## Tree basic types of "app“

**Native**

*Built specifically to the needs of the various operating systems such as Apple's iOS or Android*

**Web**

*Websites built using HTML that are designed specifically for smaller screens*

**Hybrid**

*Native app shell with feeds from the website*

# MOBILE APPS: Native APP


Written using the default language for the mobile platform, which is Objective C or Swift for iOS and Java for Android.

Compiled and executed directly on the device.

Using the platform SDK (API), the app can communicate with the platform to access device data or load data from an external website using http requests.

# MOBILE APPS: Native APP


| PROS | CONS |
| --- | --- |
| Native APIs | Language requirements |
| Performance | Not cross platform |
| Same environment | High level of effort |

# MOBILE APPS: WEB APP


Mobile websites are applications that work well on a mobile device, but are accessed through the mobile browser.

These websites viewed on a mobile device in a mobile browser, with the exception of being designed to fit a mobile device screen size.

Responsive web design can be used to make a web application - whether a conventional web site or a single-page application viewable on small screens and work well with touchscreens.

# MOBILE APPS: WEB APP


## PROS

- Maintainability
- No installation.
- Cross platform.

## CONS

- No native access
- Requires keyboard to load
- Limited user interface.

# MOBILE APPS: HYBRID APP


A hybrid app is one that combines elements of both native and Web applications

Hybrid apps are often mentioned in the context of mobile computing

For the most part, hybrid apps provide the best of both worlds

# MOBILE APPS: HYBRID APP

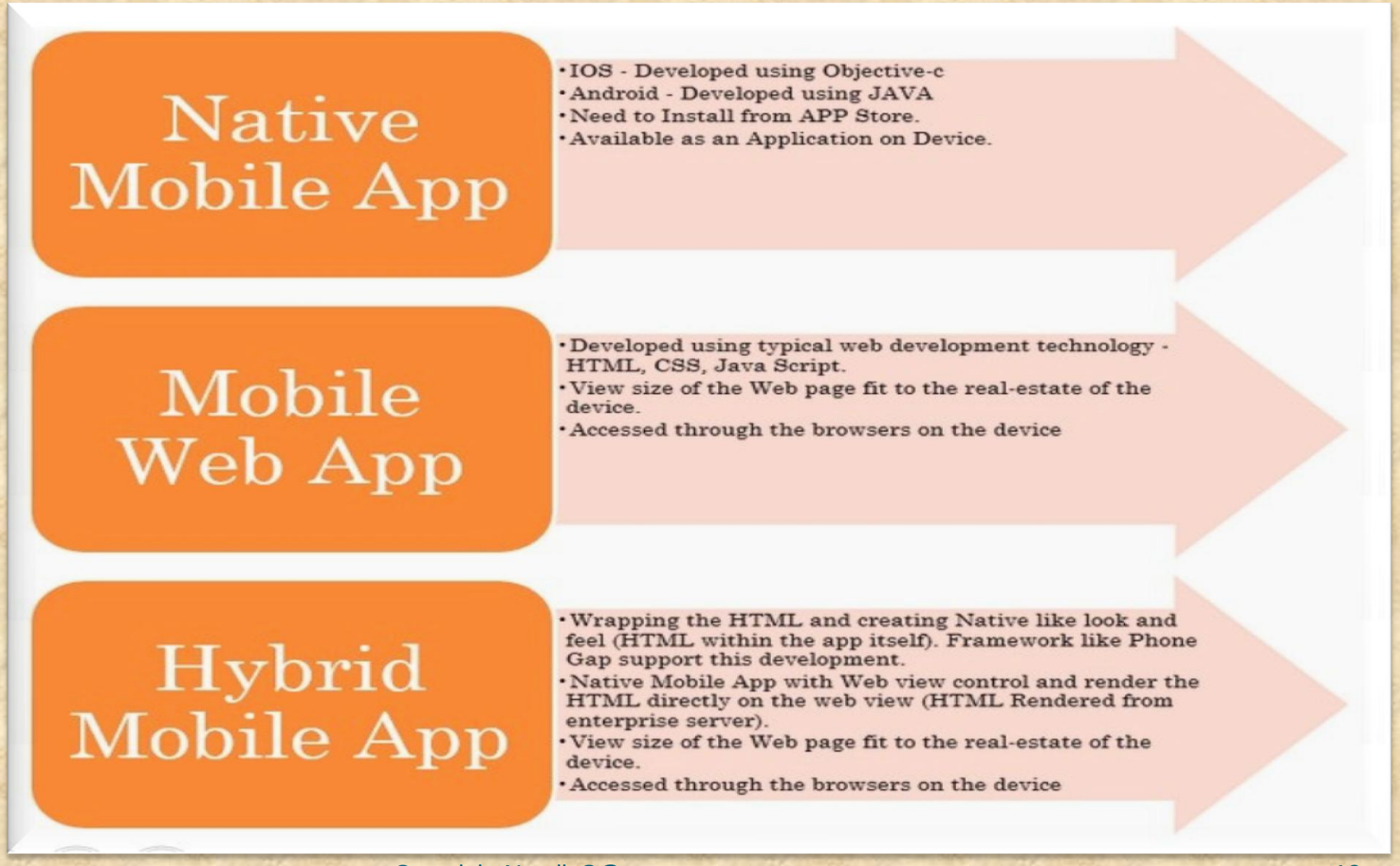

## PROS

- Cross platform
- Same skills as web development
- Access to device
- Ease of development

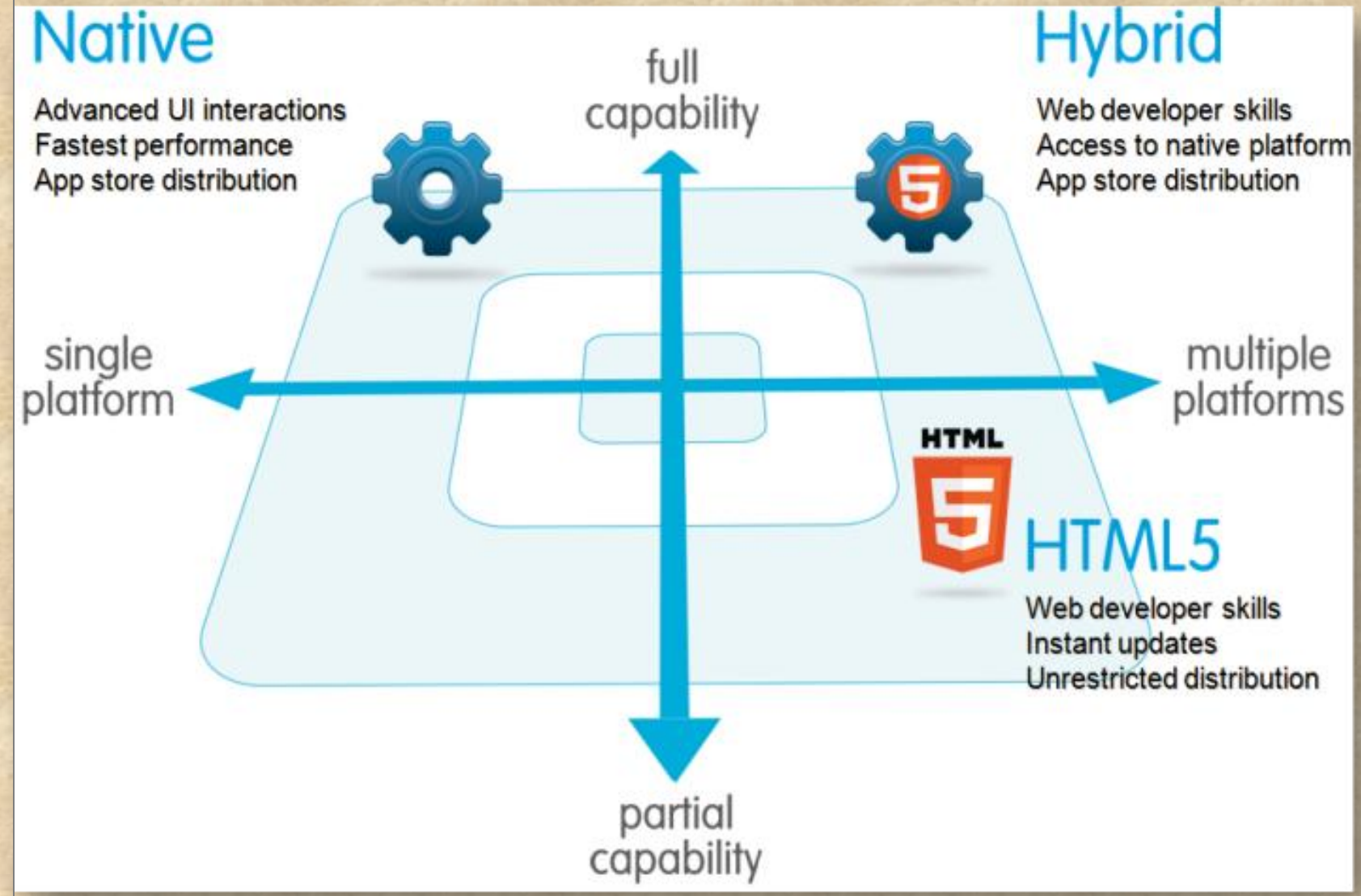
## CONS

- Web view limitations
- Native via plugins
- No native user interface controls
- Experienced developers

# MOBILE APPS: SUMMARY



# MOBILE APP types COMPARISSON



# Mobile APPS : Conclusion

| LIST | Native | HTML5 | Hybrid |
|---|---|---|---|
| **App Features** | | | |
| Graphics | *Native APIs* | *HTML, Canvas, SVG* | *HTML, Canvas, SVG* |
| Performance | *Fast* | *Slow* | *Slow* |
| Native look and feel | *Native* | *Emulated* | *Emulated* |
| Distribution | *Appstore* | *Web* | *Appstore* |
| **Device Access** | | | |
| Camera | *Yes* | *No* | *Yes* |
| Notifications | *Yes* | *No* | *Yes* |
| Contacts, calendar | *Yes* | *No* | *Yes* |
| Offline storage | *Secure file storage* | *Shared SQL* | *Secure file system, shared SQL* |
| Geolocation | *Yes* | *Yes* | *Yes* |
| **Gestures** | | | |
| Swipe | *Yes* | *Yes* | *Yes* |
| Pinch, spread | *Yes* | *No* | *Yes* |
| Connectivity | *Online and offline* | *Mostly online* | *Online and offline* |

# Overview: *Mobile APPS*

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

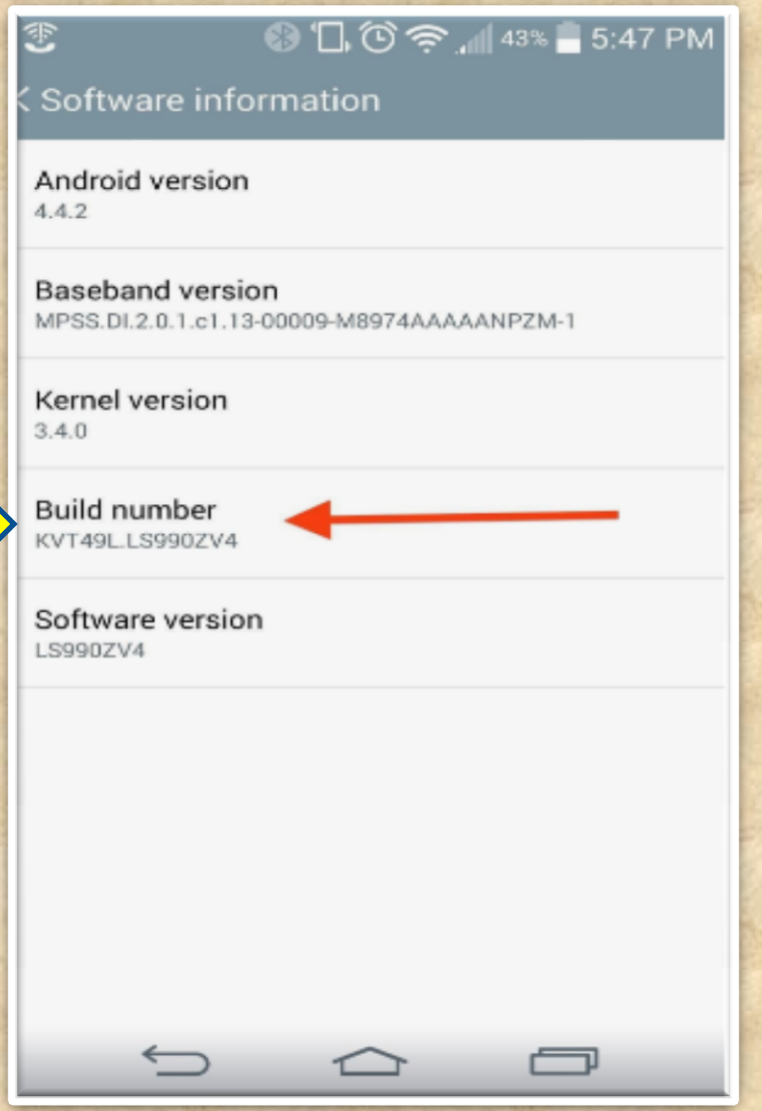
# Mobile APPS: Distribution/Installation/Logs

## How to enable Developers Options ?

1. Enable **USB debugging** in the device system settings, under **Developer options**.

2. To make it visible, go to **Settings > About phone** and tap **Build number seven times.**

3. Return to the previous screen to find **Developer options** at the bottom.



# Mobile APPS: Distribution/Installation/Logs

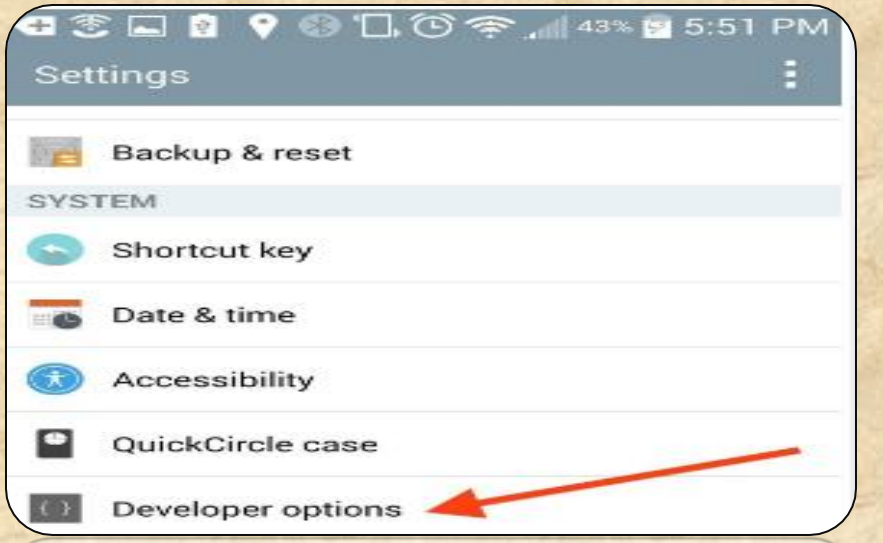
## ( contin.)How to enable Developers Options ?

Open Developers Options

Check the box **USB debugging**.

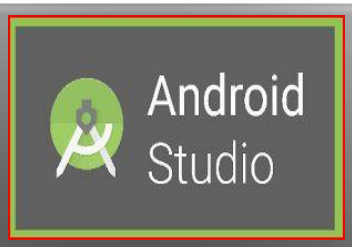
This setting will allow you to connect your device to your computer, then issue **fastboot** commands via **ADB**.

This is useful for rooting, unlocking bootloaders, installing recoveries, and a ton more.




# Mobile APPS: Distribution/Installation/Logs

# Mobile APPS: Distribution/Installation: Android .APK




What is
Android Studio?

**Android Studio** is the official integrated development environment (IDE) for **Android** platform development.

The official language for Android development is **Java**. Large parts of Android are written in **Java** and its APIs are designed to be called primarily from **Java**.

It is possible to develop C and C++ apps using the Android Native Development Kit (NDK), however it isn't something that Google promotes.

# Mobile APPS: Distribution/Installation/Logs

## What is ADB in Android Studio

Android Debug Bridge (adb) is a versatile command line tool that lets you communicate with an emulator instance or connected Android-powered device.

It is a client-server program that includes three components:

- **A client**, which sends commands. The client runs on your development machine. You can invoke a client from a shell by issuing an adb command. Other Android tools such as DDMS also create adb clients.
- **A daemon**, which runs commands on a device. The daemon runs as a background process on each emulator or device instance.
- **A server**, which manages communication between the client and the daemon. The server runs as a background process on your development machine.