<https://apimetrics.io/api-knowledge-base/apis-for-dummies/>

<https://www.youtube.com/watch?v=L4OcxA_l3ok&t=1018s>

Overall checklist for Mobile UI Testing :

* Test overall color scheme and theme of the app on the device.
* Check the style and color of icons.
* Test the look and feel of the web content across a variety of devices and network conditions.
* Test for multi-column layout – check if columns are aligned correctly and viewable even on a lower resolution.
* Test if progress indicators are visible when pages are loading.
* Check the Menus and how they are invoked.
* Check the Items contained in the Menu.
* Screen Orientation is tested in both portrait and landscape mode.
* Check the use of virtual keyboard while changing the screen mode.
* Check the pinch-to-zoom effect through touch screens and trackballs – details should not be distorted on the zooming.
* Test the sliding effect – should work in a single stroke; next screen must into the screen resolution without distortion
* Test the buttons sensitivity – should be clickable with any kind of touch (a large fingertip or stylus).
* Virtual keyboard opens up automatically when the user wants to enter text in any text field.
* Test if the application is integrated well with the mobile hard keys- start, home, menu, back buttons.
* Check if the page navigation and scrolling are working fine through the trackball.
* Test the overall responsiveness of the application on the device.

[9:12](https://pcsflex01122022.slack.com/archives/C0380UA59HR/p1650557535511879)

Overall for Usability testing :

* the buttons should have the required size and be suitable to big fingers.
* the buttons are placed in the same section of the screen to avoid confusion to the end users.
* the icons are natural and consistent with the application.
* the buttons, which have the same function should also have the same color.
* the validation for the tapping zoom-in and zoom-out facilities should be enabled.
* the keyboard input can be minimized in an appropriate manner.
* the application provides a method for going back or undoing an action, on touching the wrong item, within an acceptable duration.
* the contextual menus are not overloaded because it has to be used quickly.
* the text is kept simple and clear to be visible to the users.
* the short sentences and paragraphs are readable to the end users.
* the font size is big enough to be readable and not too big or too small.
* the application prompts the user whenever the user starts downloading a large amount of data which may be not conducive for the application performance.
* the 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.
* the application items are always synchronized according to the user actions.
* the 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

<https://developer.samsung.com/remotetestlab/deviceList.action>

<https://parsons-technology.com/how-do-i-fix-intel-haxm-is-not-installed/>

<https://www.engadget.com/apple-wallet-drivers-license-state-id-arizona-161011872.html>

**Error Guessing technique requires skilled and experienced tester. It is mainly based on intuition and experience.**

Key challenges in Mobile testing

* Too many devices globally
* Device fragmentation
* Different screen sizes
* Numerous types of mobile applications
* Mobile network bandwidth
* Agile user expectations
* Seamless user experience
* Security concerns
* Strict deadlines
* Heavy battery usage
* Too many app testing tools

**Mobile Testing vs Mobile Application Testing**

* Mobile Testing or Device Testing is a process of testing the quality of a mobile device. This type of testing includes hardware and software testing for a mobile.
* Mobile Application Testing is a process by which an application (a software developed for mobile devices) is tested for its functionality, usability, and consistency.

**Android Fragmentation**

* Android Fragmentation refers to the fact that there are a massive number of different Android OS versions available and operational in the digital world
* The Android skin - that software layer that manufacturers plaster over the top of Android.

**Difference between adb logcat and adb bugreport**

adb logcat - showing log in real time, you can filter adb logcat files. describes all the logs which has information, error, warning, fatal. contains several commands for filter. The Logcat window in Android Studio displays system messages, such as when a garbage collection occurs, and messages that you added to your app with the Log class. It displays messages in real time and keeps a history so you can view older messages.  
  
whereas adb bugreport - collect a whole info log for last session, file may be big. It has many information which is not needed for QA

This is the bug report and it contains diagnostic output for system services (dumpsys), error logs (dumpstate), and system message logs (logcat). The system messages include stack traces when the device throws an error, and messages written from all apps with the Log class.  
The bug report also contains a ​​systrace.txt file, which captures and displays the execution time of your application processes and other system processes in order to analyze the performance of the app under test.

*Bugreport also contains logcat details and other details such as diagnostic output from system services, error logs. It is available from last session. It cannot be cleared. it keeps growing depending upon the size mentioned in the settings. You can analyze performance of the app under test using systrace.txt which is a part of bugreport which includes time and processes*

**Hot Fixes / Patch**

**Patches are often temporary fixes between full releases of software packages. Patches are used to correct both large and small issues that may or may not require immediate attention, such as:**

* + **Fixing a software bug**
  + **Installing new drivers**
  + **Addressing new security vulnerabilities**
  + **Addressing software stability issues**
  + **Upgrading the software**

**Hotfixes can also solve many of the same issues as a patch, but it is applied to a “hot” system—a live system—to fix an issue:**

* **Immediately**
* **Without creating system downtimes or outages.**

**Hotfixes are also known as QFE updates, short for quick-fix engineering updates, a name that illustrates the urgency.**

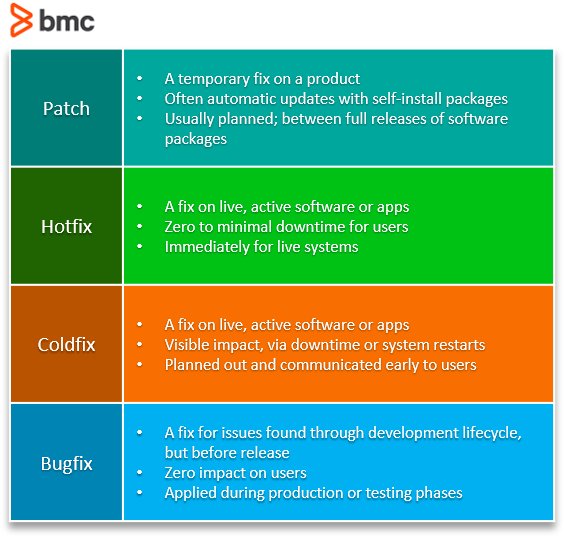
**Normally, you’ll create a hotfix quickly, as an urgent measure against issues that need to be fixed immediately and outside of your normal development flow. Unlike patches, hotfixes address very specific issues like:**

* **Adding a new feature, bug, or security fix**
* **Changing database schema**

**Importantly, hotfixes are not always publicly released, in contrast to patches.**

**ColdFix**

**Where a hotfix is executed quickly without restarting any systems or hardware, a coldfix is just the opposite. Implementing a coldfix requires users to log out of the software while entire systems need to be rebooted for fixes to go into effect.**



**Bug Leakage**

* **A bug leakage results when a bug is detected which should have been detected in earlier builds/versions of the application.**
* **A defect which exists during testing yet unfound by the tester which is eventually found by the tester/end-user is also called bug leakage.**
* **A bug release is when a particular version of s/w is released with a set of known bug(s)/defect(s). These bugs are usually low severity and/or low priority bugs. It is done when the company can afford the existence of bug in the released s/w rather than the time/cost for fixing it in that particular version. These bugs are usually mentioned in the Release Notes.**

**Release Notes**

* **A release note refers to the technical documentation produced and distributed alongside the launch of a new software product or a product update (e.g., recent changes, feature enhancements, or bug fixes). It very briefly describes a new product or succinctly details specific changes included in a product update.**

**Conditional Sign Off**

* **When the application doesn’t meet the exit criteria, the QA can do the Conditional Sign Off. For instance, when the application development exercises cross the cutoff time/deadline, and the tester has not finished the testing, they can do the Conditional Sign Off after speaking with their Test Manager.**

**OR**

* **Apart from this QA, do conditional sign off when the application does not meet the exit criteria. The conditional sign-off is ultimately a way for you to accept exiting the testing, but with a list of agreed conditions to move forward.**

**What is a QA sign off?**

* **The principle point of a QA is to ensure that the application meets the client’s needs (application should behave adequately, run smoothly, etc..) without hurting the application. The proper method of pronouncing this cycle is Sign Off. The QA can recognize/acknowledge that they have reviewed and tested the application, and now the application is prepared to deliver.**

**Limitations of Android Emulator**

**The Android Emulator doesn't include virtual hardware for the following:**

* **Bluetooth**
* **NFC (Near Field Communication - NFC is a method of wireless data transfer that allows smartphones, laptops, tablets, and other devices to share data when in close proximity. NFC technology powers contactless payments via mobile wallets like Apple Pay, Google Pay, as well as contactless cards.)**
* **SD card insert/eject**
* **Device-attached headphones**
* **USB**
* **No incoming**

**Limitations of iOS Simulator**

**It mimics only software part and not the hardware part of the devices**

**Check the Emulator after creating AVD (Android Virtual Device)**

* **emulator -list-avds**
* **emulator @avdname or emulator -avd <avdname>**
* **emulator -help**

**Enable Developer Options in Android**

* **Tap the Build Number option 7 times. - Settings > About Phone > Build Number**
* **Toggle the ON option in the Developer Options menu. - Settings > System > Advanced > Developer Options**

**Enable USB Debugging**

* **Connect Android device to your computer using a USB cable**
* **Toggle the USB debugging option in the Developer Options menu (Allow USB debugging)**

**Connect through WIFI**

1. **Connect using USB Cable**
2. **Assign a port using command – adb tcpip 5555**
3. **Disconnect from target device**
4. **Make sure both are in same WiFi**
5. **Check the ip address of the phone and use the following command to connect**

**adb connect <ip address>:portnumber**

**What is adb – Android Debug Bridge is a tool to communicate with device like installing apps, debugging , access to unix shell for different operations**

**Components of adb:**

* **A client, which sends commands. The client runs on your development machine. You can invoke a client from a command-line terminal by issuing an adb command.**
* **A daemon (adbd), which runs commands on a device. The daemon runs as a background process on each device.**
* **A server, which manages communication between the client and the daemon. The server runs as a background process on your development machine.**

**Adb commands**

* **adb devices**
* **adb install <path to apk>**
* **adb [options]**
  + - **-d 🡪 device**
    - **-e 🡪 emulator**
    - **-s 🡪 Serial number of device or emulator**
* **adb [options] install [options] <path\_to\_apk>**
* **adb [options] install [-r] <path\_to\_apk> (for updating app)**
* **adb [options] uninstall [options] <package\_name> 🡪 To uninstall use package name**
* **To go into Shell -- > adb [options] shell**
* **adb [options] shell <shell\_command*>***
* **adb [options] shell pm list packages -f <keyword> 🡪 adb -e shell pm list packages -f bus**
* **adb -e shell pm list packages -3 🡪Displays only third party apps**
* **adb kill-server**
* **adb start-server**
* **adb shell dumpsys package <packagename> | grep “versionCode” 🡪 returns the min Android version supported and max Android version supported**
* **“adb shell dumpsys battery – change battery level , adb shell dumpsys battery set level 5 🡪 for testing low battery in androis , adb shell dumpsys battery unplug 🡪 device appears as unplugged , adb shell dumpsys battery reset 🡪 resets all the battery configs**
* **adb push <path to filename> /sdcard/ 🡪 Push files to device – Often times you need to push files or folders to the device**

**adb logcat Commands**

* **adb logcat –help**
* **Logcat is a command-line tool that dumps a log of system messages, including stack traces when the device throws an error and messages that you have written from your app with the**[**Log**](https://developer.android.com/reference/android/util/Log)**class**
* **[adb] logcat [<option>] ... [<filter-spec>] ...**
  + **Mac: adb logcat | grep -i bus**
  + **Win: adb logcat | find /i “bus”**
* **Win: adb logcat | find /i “bus” > LogBus.txt 🡪 to write log into file**
* **Mac: adb logcat|grep –i bus > LogBus.txt**
* **To capture Crash logs – use FATAL or Error in logs and Exception like Nullpointer, Arithmetic etc**

**Application Not Responding (ANR)**

* **"Application Not Responding" (ANR) error is triggered when UI thread of Android app is blocked for too long.**
* **If the app is in the foreground, the system displays the ANR dialog to the user. The ANR dialog gives the user the opportunity to force quit the app.**
* **Android stores trace information when it experiences an ANR.**

**there are multiple /data/anr/anr\_\*files**

**How to use adb as root**

* **adb root   
  adbd will be restarted as a root $**
* **adb unroot  
  adbd will be restarted as non root #**

**Retrieving ANR reports from device to your system**

**Use adb [options] if multiple devices are connected**

1. **adb shell ls /data/anr   
   List all ANRs from emulator (device)**
2. **adb pull /data/anr/<filename>  
   Pull a particular ANR file to your machine or**
3. **adb pull /data/anr  
   Pull all folder to your machine**

**Android Bug Report**

* **A bug report contains device logs, stack traces, and other diagnostic information to help you find and fix bugs in your app.**

**Capture Bug Report from device**

**Developer options 🡪 Bug Report 🡪 Select type of Report 🡪 tap Report. After few moment, you will get notification then you can share / copy the report by tapping notification.**

**Capture Bug Report From Emulator**

**Click More in Extended panel 🡪 In extended window select Bug Report , wait for that to finish and save report**

**Capture Bug Report using adb command**

* **adb bugreport**

**Logcat priority**

* **The priority is one of the following character values, ordered from lowest to highest priority:**
  + **V: Verbose (lowest priority)**
  + **D: Debug**
  + **I: Info**
  + **W: Warning**
  + **E: Error**
  + **F: Fatal**
  + **S: Silent (highest priority, on which nothing is ever printed)**

**Filtering logs by priority**

* **adb logcat \*:W 🡪 Shows all warnings (W) and all other logs with higher priority**
* **adb logcat OkHttp:D \*:S 🡪 Shows all debug (D) records with tag OkHttp, the rest stays silent**

**Logcat Format Modifiers**

* **[adb] logcat [-v <format>]**
* **adb logcat -v color -d  
  Shows each priority level with a different color**
* **adb logcat -v brief**

**Display priority, tag, and PID of the process**

**Screen Shot**

**From device : - you can capture a screenshot with a key-combination: Simultaneously press-and-hold Power and Volume-down.**

* **adb shell screencap /sdcard/<file\_name>.png** 
  + **Take a screenshot and save it on the emulator or device**
* **adb pull /sdcard/<file\_name>.png <destination\_path>**
  + **Pull the file from emulator or device**

**Screen Recording**

* **adb shell screenrecord /sdcard/<file\_name>.mp4**
* **adb pull /sdcard/<file\_name>.mp4 <destination\_path>**

**Signed apk 🡪 Application signing ensures that one application cannot access any other application except through well-defined IPC when installed in the device.**

**How do you identify that it is a hybrid application?**

Go to Developer Options from Settings and turn on Show Layout bounds. And then launch your app and see the layout highlighted. If there are lots of rectangles highlighted, it's native. Otherwise, hybrid.

**Make calls and send text**

**adb shell am start -a android.intent.action.CALL -d** [**tel:+1<Phone**](tel:+1%3cPhone) **number>**

**adb shell am start -a android.intent.action.SENDTO -d sms:+1<phonenumber> --es sms\_body "Test" --ez exit\_on\_sent false”**

**List OneThing Which You Cannot Do With Emulators But You Can Do With A Real Device.**

You can test

* Memory card mount/unmount scenarios
* Actual performance of your application
* Bluetooth related testing.
* Cellular network with wifi / without wifi

**Difference between Selenium and Appium.**

Selenium is an open-source tool that allows automation for Web applications run in Browsers. Appium is used for automating mobile applications and web applications which run on mobile devices.

**XCODE**

**Xcode is the developer toolset**

**How to get application for testing on device?**

* **App Store**
* **TestFlight**
* **Build from Xcode to connected device/simulator**
* **Create and distribute .ipa file**

**Simulator**

* **Simulator runs on your Mac and behaves like a standard Mac app while simulating iPhone, iPad, Apple Watch, or Apple TV environments.**
* **Each simulator has its own simulation environment, independent of the others, with its own settings and files.**
* **Multiple simulators can be run at the same time.**

**What can you do with simulator**

* **Interact with your apps on iOS, watchOS, and tvOS using your pointer and keyboard.**
* **Prototype and debug your apps.**
* **Optimize your graphics.**
* **Test your apps.**

**Starting Simulator**

* **Xcode. Start the simulator  
  Xcode -> Open Developer Tool -> Simulator**
* **Terminal  
  $ xcrun simctl boot <device>**

**Where to check logs ?**

* **Xcode**
* **On Mac (for simulators) ~/Library/Logs/CoreSimulator/<simulator-hash>/system.log**
* **Console (for all devices)**
* **On device**

**Command Line to control simulators**

* **$ xcrun simctl help**
* **$ xcrun simctl list  
  Show a list of all simulators**
* **$ xcrun simctl boot <device UID>**
* **Start a simulator**
* **$ xcrun simctl list | grep Booted**
* **Show a list of booted simulators**
* **$ xcrun simctl install booted ~/Library/Developer/Xcode/DerivedData /<app bundle>/Build/Products/Debug-iphonesimulator/<name>.app  
  Install builded application**
* **$ xcrun simctl launch booted <bundle\_ID>  
  Launch an app**
* **$ xcrun simctl uninstall booted <bundle\_ID>  
  Uninstall an app**
* **$ xcrun simctl spawn booted log stream  
   Prints a log data to the terminal from a booted device**
* **$ xcrun simctl spawn booted log stream --style=json  
  Prints a log data to the terminal in JSON format**
* **$ xcrun simctl spawn booted log stream > file\_name.log  
  Creates and save the log output in file\_name.log**
* **$ xcrun simctl spawn booted log stream --predicate 'eventMessage contains "<bundle\_ID>"'  
  Shows logs from a specific app**
* **$ xcrun simctl erase <device UID>  
  Erase the content of simulator**
* **$ xcrun simctl erase all  
  Erase the content on all simulators**
* **$ xcrun simctl shutdown <device UID>  
  Shuts down simulator**
* **$ xcrun simctl io booted screenshot <name>.png**
* **Screenshot**
* **$ xcrun simctl io booted recordVideo <name>.mp4  
  Video Recording**
* **$ xcrun simctl openurl booted "URL"  
  Opens url in the default browser**
* **xcrun simctl install booted <unique deviceid> <path to .app> 🡪 If more than one simulator is running use unique device id to install / uninstall/ other operations**

**Emulator vs Simulator**

* **Android Emulator. It duplicates every aspect of the original device’s behavior, both hardware and software.**
* **Apple Simulator. It simulates devices OS, but it doesn’t attempt to simulate the real device’s hardware.**

**Emulator :**

* **Advantages**
  + **Can simulate both software and hardware**
  + **Run the code without any modification**
  + **Help you find unexpected behavior**
* **Disadvantages**
  + **Very slow**
  + **Hard to configure and maintain**
  + **Don’t take into consideration factors like battery overheating or conflicts with other (default) apps**

**Simulator :**

* **Advantages**
  + **Fast**
  + **No setup needed (or minimum involvement)**
  + **Can be used to study the behavior of an app**
  + **Help you find unexpected behavior.**
* **Disadvantages**
  + **Can only simulate the software part**
  + **Apps may might need a modified code to run on simulators**
  + **Simulation results may be difficult to analyze, due to incomplete data**

**Real Devices**

* **Advantages**
  + **Provide real environment**
  + **Use all hardware like GPS, network, Bluetooth and many others in real time**
  + **Perform faster than emulators and simulators**
  + **Allow to use sensors like orientation, gyroscope and others**
  + **Can catch battery drain, excessive usage of CPU, GPU and RAM**
* **Disadvantages**
  + **Testing on multiple models is not only frustrating and time consuming but also expensive**
  + **The resources spent on the maintenance of the real devices**
  + **The devices may not be released yet**

**Device Farms**

* **Advantages**
  + **Run tests (including automated test) and interact with a large selection of physical devices.**
  + **Simulate real-world environments**
  + **Cheaper than buying devices (depending on how many tests we make and how many devices we run it on)**
* **Disadvantages**
  + **Limited to scripted automated tests**
  + **Cannot use device camera**
  + **Cannot use carrier networks**
  + **No GPS testing**
  + **Cannot “adb” into devices**
  + **Cannot physically see or feel how the app performs on each device.**

**Native Mobile Applications**

* **The application that was developed for specific platform using native tools**
* Twitter, or games, such as Pokémon GO., Google maps, pinterst, Spotify

**Advantages**

* **Fast and Responsive (reliable and most responsive experience to users**
* **Full Mobile Device Feature Accessibility (hardware, software)**
* **Match app UI/UX to platform conventions**
* **Offline Availability**

**Disadvantages**

* **More than one codebase**
* **Native Apps cost more**
* **Native Apps take longer to build usually**

**Mobile Web Apps**

* **Mobile Web applications – the applications for mobile devices that require only a Web browser to access those apps.**
* **They use Web technologies and are not limited to the underlying platform for deployment.**
* Starbucks, spotify , Uber, make my trip

**Advantages**

* **Compatibility**
* **Upgradability**
* **Findability**
* **Shareability**
* **Support and Maintenance**

**Disadvantages**

* **No Offline Availability**
* **Limited Mobile Device Feature Accessibility**
* **Stability (saving the data)**
* **UI guidelines not respected**
* **No App Store or Google Play Access**

**Remote Debug Android devices**

**Connect your device using usb**

**chrome://inspect#devices**

**Hybrid Mobile Applications**

* **Hybrid applications are web applications (or web pages) that are wrapped in a native application**
* **They need to be installed, run on device itself, and use device’s browser engine (but not the browser) to render the HTML and process the JavaScript locally**
* Instagram, discord, gmail,
* **Advantages**
  + **One codebase to manage (for web-part)**
  + **You save time and money**
  + **Easier to scale**
  + **You still have access to device features**
* **Disadvantages**
  + **Performance Challenges**
  + **The UX of the app may suffer**
  + **Could be slow**

**Cross-Platform Mobile Applications**

* **Cross-platform mobile applications are mobile apps developed to function for multiple mobile platforms , with more focus on code reusability🡪 writing code to use it in multiple os/platforms**
* **These apps are compatible with more than one operating system, such as iOS and Android**
* Slack, Skype, Bloomberg
* **Advantages**
  + **Reusable Code Components. One codebase to manage**
  + **You save time and money**
  + **Speed of development**
  + **Reduced Costs**
* **Disadvantages**
  + **Performance Challenges**
  + **Integration challenges**
  + **Limited User Experience**
  + **Cross-platform development is tough**