1. Android has four base level components: 1. Activities which dictate the UI and handle the user interaction to the device screen. An activity represents a single screen in which actions are performed on. 2. Services which handle background processing within an application. A service might play music in the background while a user is in another application. 3. Broadcast Receivers handle the communication between the Android OS and applications. To put it simply they respond to messages from other applications or the system. 4. Content Providers handle data and database management issues. A content provider supplies data from one application to others on request. These requests are handled by the ContentResolver class.
2. The activity life cycle consists of onCreate which is called when the activity is first brought to life. onStart means the activity is now visible but not ready for user interaction yet. This life cycle comes in handy to register a BroadcastReceiver to monitor any changes affecting the UI. onResume means the activity is in the foreground and ready for user interaction. Usually used for starting animations and exclusive access devices. onPause the activity is about to go into the background and has ceased user interactions. This may happen when another activity is launched in front of the current activity. onStop is when the activity is no longer visible to the user. onRestart is called when the activity has been stopped before it Is started again, it is rare to use this callback. onDestroy is the counterpart to onCreate, usually triggered when the activity calls finish() or the system needs to allocate memory.
3. The first step of the sprint usually held on Monday is the sprint planning and first stand up. During the week other standups are held where a discussion of what I worked on previously, currently, and in the future. Any issues need to be addressed that will hinder my progress. We also go over story points, which can be considered in the estimated completion time. On the Friday of the first week is when the first grooming session is held. This is where we discuss again any constraints or issues that will prohibit completion of stories in a timely manner. The Wednesday of the second week is when the first pull request is submitted. Generally, your code is overlooked by a senior developer or a leader on a project. Any optimization suggestions are generally made at this time. The Thursday following the first pull request is when the demo takes place. This is usually when you check if the UI is up to standard and marketable. The Friday of generally the second week or whenever the sprint stage ends consists of a standup and the next pull request is submitted and the final code check is done.
4. ART is the managed android runtime used by applications and some system services on android. ART and its predecessor Dalvik were originally created specifically for the android project. ART as the runtime executes the Dalvik Executable format and Dex bytecode specification. They are both compatible runtimes running Dex bytecode, so apps developed for Dalvik most likely will work when running with ART.
5. Android is structured in the form of a software stack comprising applications, an operating system, run-time environment, middleware, services and libraries. Each layer of the stack, and the corresponding elements within each layer, are tightly integrated and carefully tuned to provide the optimal application development and execution environment for mobile devices. Starting from the bottom, the android software stack consists of the Linux Kernel which provides a level of abstraction between the device hardware and the upper layers of the stack. The Linux Kernel is the only kernel used in android. Android studio takes apps built within it and compiles them into an intermediate bytecode format known as Dex format. When the app is loaded onto the device, the ART uses a process referred to as Ahead-of-Time (AOT) compilation to translate the bytecode down to the native instructions required by the device processor. This is known as Executable and and Linkable format (ELF). AT each launch the ELF version is run resulting in faster performance and improved battery life. In addition to the standard Java development libraries, Android also has a set of Java based libraries that are android specific. These libraries consist of the application framework libraries and those that facilitate interface building, graphics drawing, and database access. Though the android core libraries are Java based, most of these core libraries are in fact wrappers around of set of C based libraries. These C/C++ libraries are used to include a wide range of functions to work with the underlying linux kernel. Typically, the android developer will access these libraries through the java based Android core libraries. They can directly access libraries in the kernel using the Android Native Development Kit (NDK) which can call non android native languages within java code using the Java Native Interface (JNI). The application framework is asset of services that collectively form the environment for running and managing Android apps. This framework implements the concept that Android apps are constructed from reusable, interchangeable, and replaceable components. At the top of the stack is where the applications are located. These consist of both native android apps as well as third party apps installed by the user.