



# How to Train your Dragon Android

Sachin Murali . G  
03-10-2020 for GECWAA

# Topics

---

- Android Software Stack and Development Intro**
- Source Code Management and Collaboration**
- Simple Calculator Application**
- Try it yourself - ToDo list**
- Q&A**

# Getting Started

- 1. Download and install Android Studio**
- 2. Create New Project**
- 3. Run Hello World Code in “Android Virtual Machine”**
- 4. Create a new Repo in GitHub**
- 5. Push a new commit onto GitHub**

## Java CLI App code

```
1 package me.sachinmuralig;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         // the code goes here
7     }
8 }
```

## Android Hello World Code

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The project is named "Simple\_Calculator". It contains an "app" module with a "manifests" folder containing "AndroidManifest.xml", a "java" folder containing "me.sachinmuralig.simplecalculator" package with "MainActivity.java", and a "res" folder with "drawable", "layout", "mipmap", and "values" subfolders.
- Main Activity Java Code:** The code for "MainActivity.java" is displayed in the right-hand editor. It imports AppCompatActivity and sets the content view to "activity\_main".
- Gradle Scripts:** The build configuration files "build.gradle" (Project and Module), "gradle-wrapper.properties", "proguard-rules.pro", "gradle.properties", "settings.gradle", and "local.properties" are listed.
- Bottom Navigation:** The navigation bar includes tabs for Version Control, Terminal, Build, Logcat, and TODO, along with a message indicating the daemon started successfully 13 minutes ago.

**View Mode**

**Flavours**

The screenshot shows the Android Studio interface with a project named "Simple\_Calculator". The left sidebar displays the project structure:

- app**:
  - manifests**: Contains `AndroidManifest.xml`.
  - java**: Contains a package `me.sachinmuralig.simplecalculator` with a class `MainActivity`.
  - res**: Contains **drawable**, **layout**, **mipmap**, and **values** folders.
  - Gradle Scripts**: Contains files like `build.gradle`, `gradle-wrapper.properties`, and `local.properties`.
- 1: Project**
- 2: Favorites**
- 3: Structure**

The main area shows the `MainActivity.java` file:

```
1 package me.sachinmuralig.simplecalculator;  
2  
3 import android.os.Bundle;  
4  
5 public class MainActivity extends AppCompatActivity {  
6     @Override  
7     protected void onCreate(Bundle savedInstanceState) {  
8         super.onCreate(savedInstanceState);  
9         setContentView(R.layout.activity_main);  
10    }  
11}
```

Annotations with arrows point to specific parts of the interface:

- An arrow from the "View Mode" label points to the "Resource Manager" tab in the sidebar.
- A callout bubble from the "Flavours" label points to the "Gradle Scripts" section in the sidebar.
- A callout bubble from the "Device Logs" label points to the bottom navigation bar.
- A callout bubble from the "Build scripts. Eg: Dependencies, flavours etc" label points to the "local.properties" file in the "Gradle Scripts" section.
- A callout bubble from the "Constants and other values goes in this folder" label points to the "values" folder in the project structure.
- A callout bubble from the "Screen layouts goes in layout folder" label points to the "layout" folder in the project structure.
- A callout bubble from the "Images used in your app goes in this folder" label points to the "drawable" folder in the project structure.
- A callout bubble from the "The Java Code Goes Here" label points to the `MainActivity.java` file in the code editor.
- A callout bubble from the "Everything the OS should know about your app goes here" label points to the `AndroidManifest.xml` file in the project structure.

**Everything the OS should know about your app goes here**

**The Java Code Goes Here**

**Images used in your app goes in this folder**

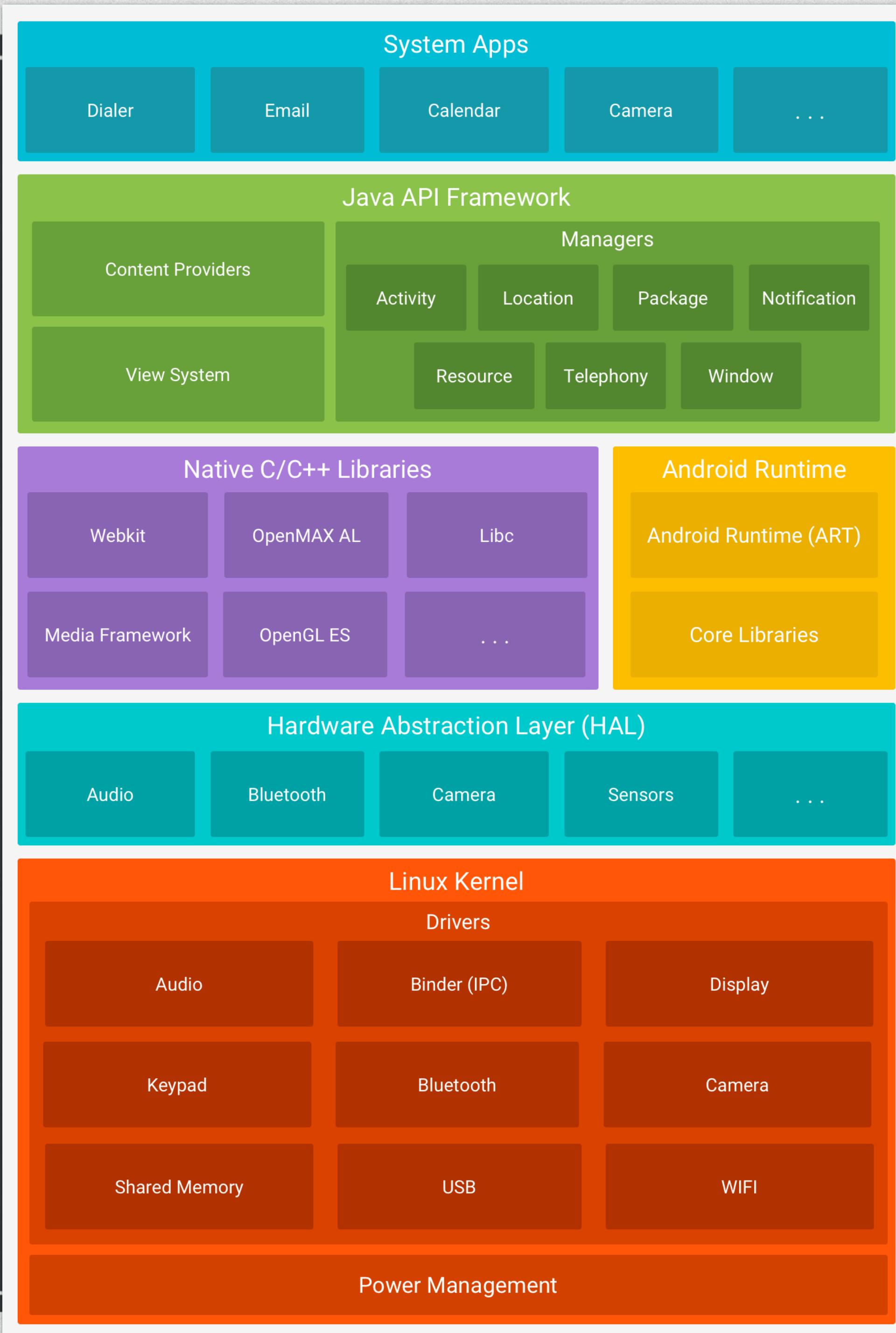
**Screen layouts goes in layout folder**

**Constants and other values goes in this folder**

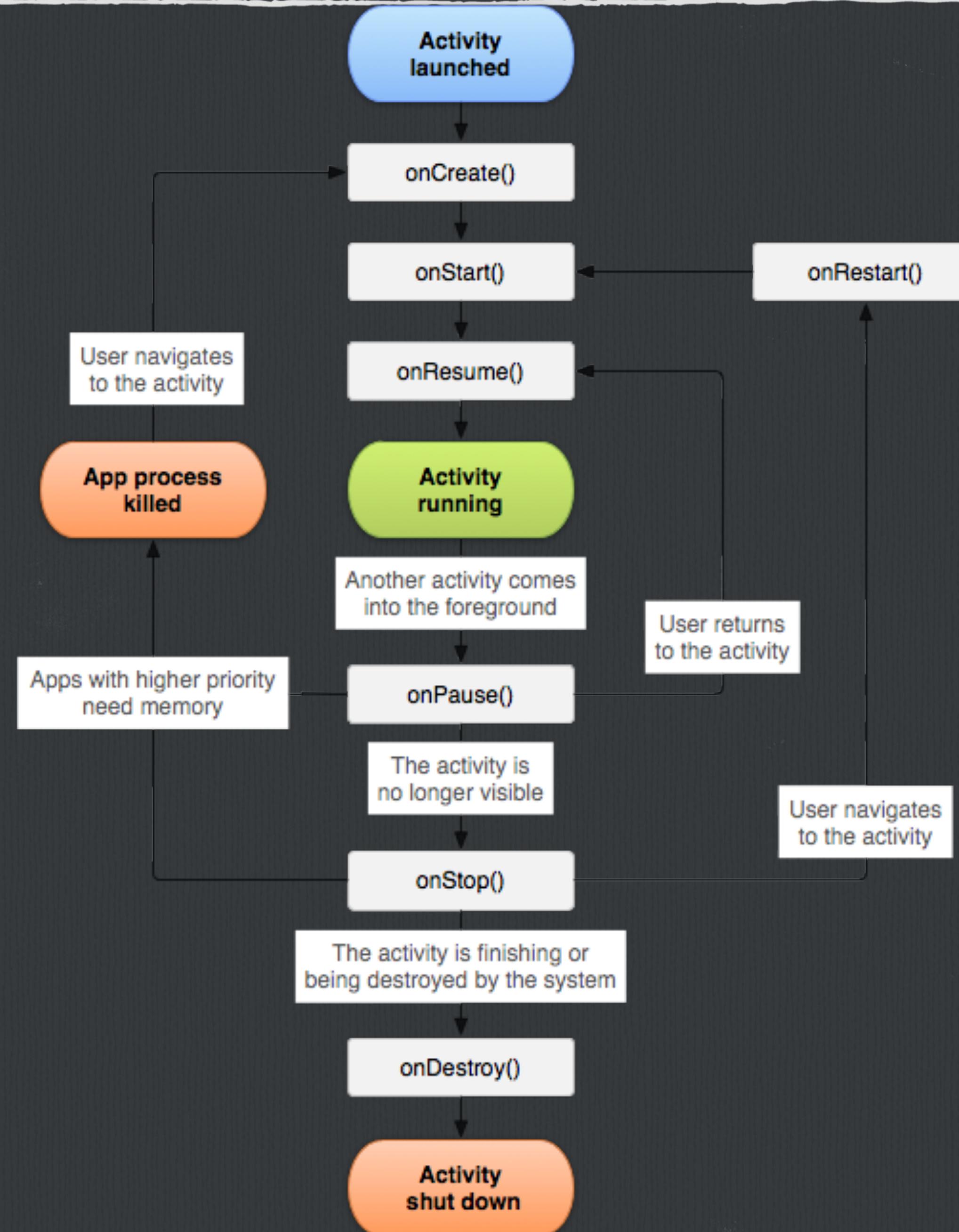
**Build scripts. Eg: Dependencies, flavours etc**

**Device Logs**

# Android Software Stack



# Activity Lifecycle



Into the nitty gritty of code

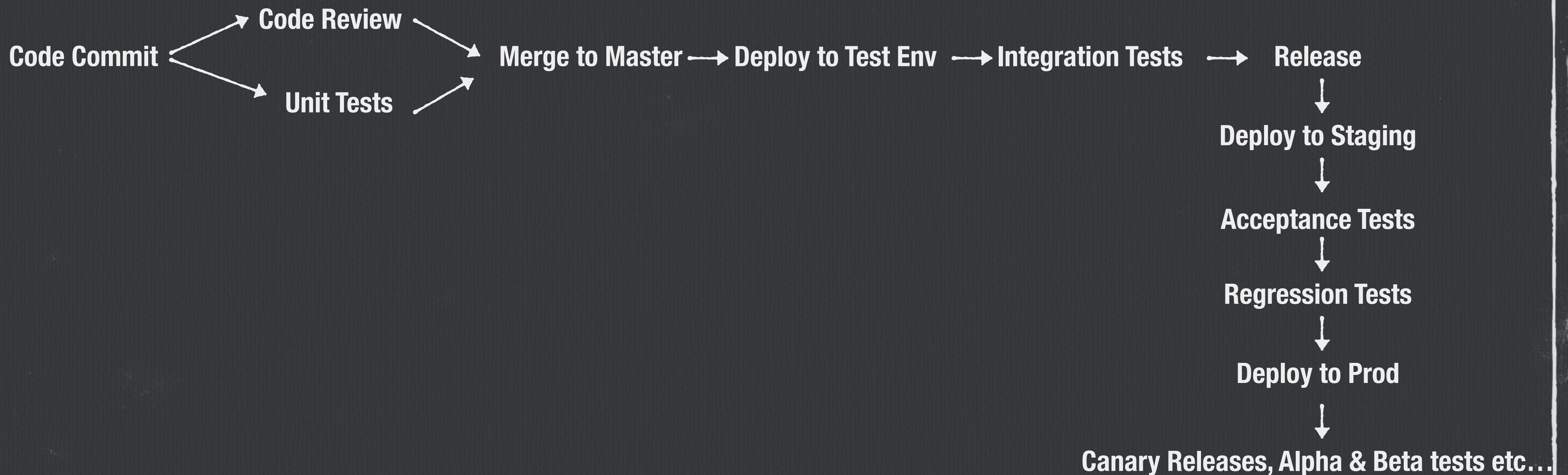
**“A Code that cannot be tested is flawed”**

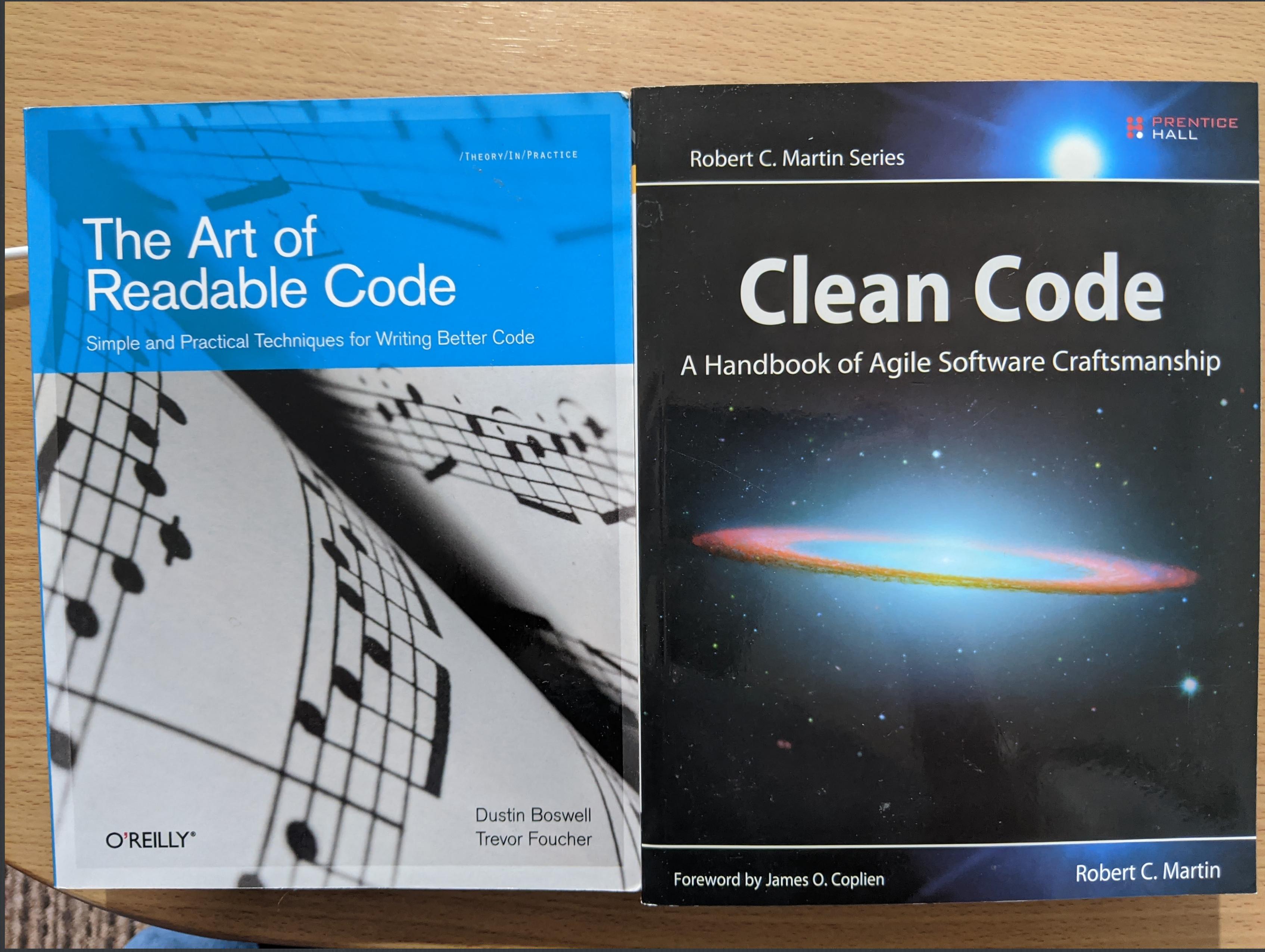
# Publishing Your App

---

- <https://developer.android.com/studio/publish>
- Continuous Integration**  
Merge changes to master branch as often as possible
- Continuous Delivery**  
Release the changes quickly in a sustainable way
- Continuous Deployment**  
Deploy code to production environment automatically
- Together coined as CI/CD/CD**

# Typical Release Pipeline





# Try it yourself

- Clone <https://github.com/sachinmuralig/simplecalculator>
  - Try running it in your own device
  - Try making it look like a handheld calculator
- Build A TODO list App
  - Use XAMPP to setup a server to save TODO list



# Q&A