Unit 2:

Development Process for Mobile Applications

Mobile Application Development (MAD)
CCIT 4059, 2025-2026

Outline

- Mobile App Development Process
  - Initiation, Requirement and Planning
  - Design
  - Implementation
  - Evaluation and Distribution (Deployment)
  - Operation and Maintenance
- Operational Environment of Mobile Applications
- Hardware and Software Constraints
- Developing Android App
  - Typical Workflow of Developing App

Mobile App Development Process

- Creating a mobile app is similar to the of typical software development process
- However, the development cycle for mobile apps is often much shortened
  - Design and produce without delay
  - Rush to launch into the market
  - Get feedbacks as soon as possible
  - Modify right away if needed, else simply scrap it and re-develop another one
- Development process of a mobile application project involves 5 stages:
  1) **Initiation, Requirement and Planning**
  2) **Design**
  3) **Implementation**
  4) **Evaluation and Distribution (Deployment)**
  5) **Operation and Maintenance**

Mobile App Development Process (Initiation, Requirement and Planning Stage)

1) Initiation Stage:

1.1 Objectives of Initiation

- Identify the problem the app will solve
  - Background study and research are required
- Define target users and use cases
- Gather functional and non-functional requirements
- Plan resources, timeline, risks, and feasibility

1.2 Requirement Gathering

- **Functional requirements:** What the system *should* do
  - e.g., user login, GPS tracking, in-app purchases
- **Non-functional requirements:** Quality constraints
  - e.g., performance, security, usability, compatibility

Mobile App Development Process (Initiation, Requirement and Planning Stage)

1.3 Planning Activities

- Define project scope
- Determine app features (feature list / user stories)
- Estimate time, cost, resources
  - A detailed plan with schedule (Start to End)
- Risk identification and mitigation planning
- Selecting technology stack (native, hybrid, cross-platform)

- Design ideas and concepts should be created towards the end of this planning stage,
  - if possible, where project objectives and requirements are refined
  - Detailed plan and schedule for managing the whole project is needed

Mobile App Development Process (Initiation, Requirement and Planning Stage)

Self-learning

- Questions to be answered in early stage 1:
  - Main goals and objectives of the mobile app?
    - Why is such a project and mobile app needed?
    - What does the client (or boss) really want?
  - Whom is the product interacting with?
    - Who are the target end-users, the markets?
  - When, where and how is the product used?
    - Operational conditions and environments?
  - Resources and supports available?
    - Manpower, budget, time
    - Development technology and platform
    - Others (e.g., Fund raising)

Mobile App Development Process (Initiation, Requirement and Planning Stage)

Self-learning

- Background study and serious research work are required, esp. in large projects
  - Research to understand the client & project
    - A general background study
    - Client and expected end users
    - Markets, competitors, laws & regulations
  - Research sources
    - Company salespeople, showroom
    - Visit competitors, compare products & services
    - Internet, surveys & interviews

Mobile App Development Process (Initiation, Requirement and Planning Stage)

- Functional requirements define *what* a system should do, while non-functional requirements define *how* the system should perform.
  - Both are essential for delivering a complete, high-quality product.

- Functional requirements
  - describe specific features, tasks, or operations that the system must perform to meet business and user needs.
  - They are directly testable and visible to users.

- Non-functional requirements (NFRs)
  - focus on quality attributes like performance, security, usability, and scalability, ensuring the system operates efficiently and reliably under various conditions.

Mobile App Development Process (Design Stage)

2) Design Stage

- Good product first comes with good design (creativity, innovative and usability of the apps)
  - Create UI/UX wireframes or mockups
  - Define navigation flow, layout, and visual style
  - Architectural design (e.g. Model-View-Controller, APIs, database)
- Visual design and user interface (human-computer interaction) design
- Important to meet identified requirements and to focus on the actual user
  - “user-centered” design approach

Mobile App Development Process (Design Stage)

- UI (User Interface) and UX (User Experience).
  - UI focuses on the visual and interactive elements, while UX emphasizes the overall user journey and experience.

UI/UX Design

- Wireframes → Mockups → Prototypes
- Key design principles: consistency, emphasis, balance, contrast, proportion, movement, white space, accessibility
- Navigation structures: tab bars, buttons, drawers, Fragment, bottom navigation, gesture-based navigation, etc.

Mobile App Development Process (Design Stage)

Architectural Design

- Common mobile app architectures:
  - MVC (Model–View–Controller)
  - MVP (Model–View–Presenter)
- Data design: local storage, database, cloud sync., API integration
- Component layout and screen flow diagram
  - Linear layout

Mobile App Development Process (Implementation Stage)

3) Implementation

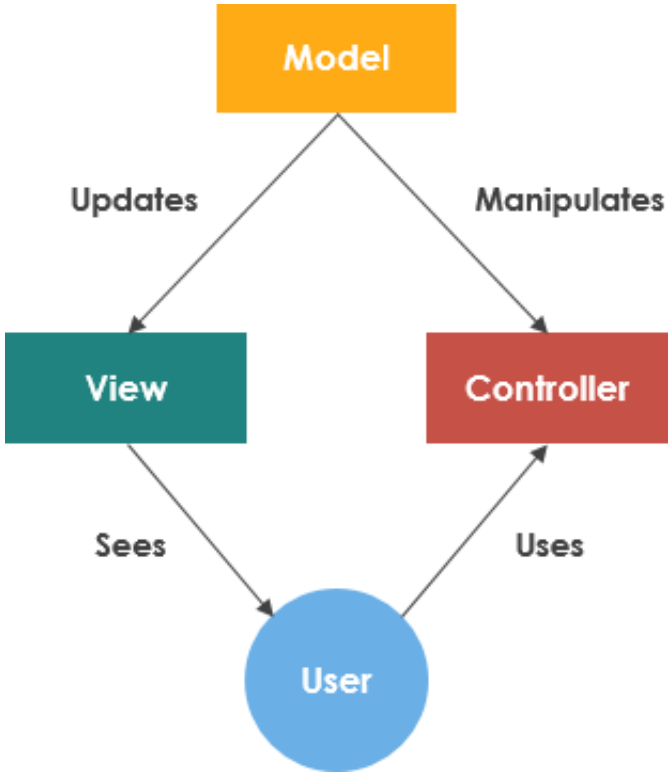
Development Activities

- Setup development environment (Android Studio / Xcode)
- Implement UI components and interactions
- Backend development (if applicable): authentication, databases, APIs
- Integration of services: maps, notifications, analytics, payments

- For Android apps,
  - Android SDK is needed – the software development kit developed by Google, mainly for developing native applications of Android with Java language

MVC (Model View Controller)

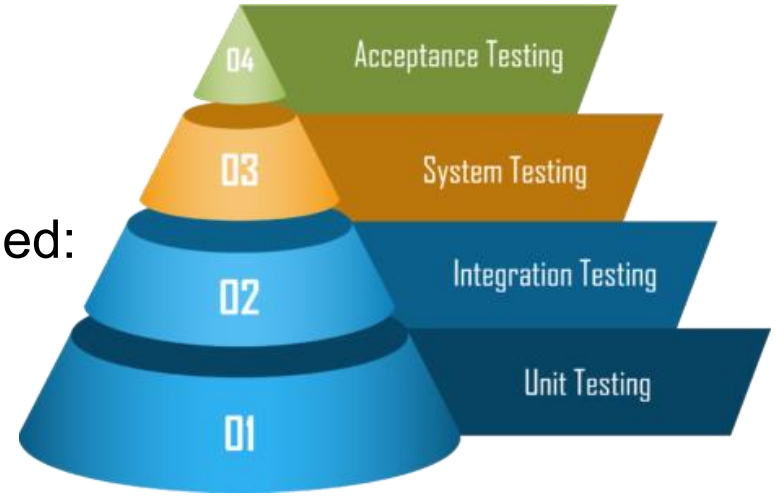
- MVC helps structure the separate the concerns of an application into three parts
- Model is responsible for managing the data of the application. It receives user input from the controller.
- View means the presentation of the model in a particular format.
- Controller responds to the user input and performs interactions on the data model objects. The controller receives the input, optionally validates it and then passes the input to the model.



MVC Framework

Mobile App Development Process (Implementation Stage)

- **Testing During Development**
  - Unit testing
  - Continuous integration (CI)



- Comprehensive software testing is needed:
  - Meets the requirements
  - Works as expected
  - Satisfies the needs of stakeholders
  - Can be implemented with the same characteristics
- Levels of software testing:
  - Unit testing, Integration testing, System testing, Acceptance testing

Mobile App Development Process (Implementation Stage)

- Unit testing level:
  - Individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use
  - This testing unit could be viewed as the smallest testable part of an application

Mobile App Development Process (Implementation Stage)

- Integration testing level:
  - Individual modules combined and tested as a group
  - It occurs after unit testing
  - Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing

Mobile App Development Process (Implementation Stage)

- System testing level:
  - Conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements
  - System testing generally falls within the scope of black box testing, and should require no knowledge of the inner design of the code or logic
  - System testing takes all of the "integrated" components that have passed integration testing and also the software system itself integrated with any applicable hardware system

Mobile App Development Process (Implementation Stage)

- Acceptance testing level:
  - Conducted to determine if the requirements of a specification or contract are met
  - It may involve performance tests
  - Software developers often distinguish acceptance testing by the system provider from acceptance testing by the customer (the user or client) prior to accepting transfer of ownership
    - Acceptance testing performed by the customer is also known as user acceptance testing (UAT), end-user testing, site (acceptance) testing, or field (acceptance) testing

Mobile App Development Process (Implementation Stage)

- Software testing “box” approaches:
  - Black-box testing treats the software as a "black box", examining functionality without any knowledge of internal implementation
  - White-box testing: tests internal structures or workings of a program (also known as clear box testing)

- Towards completion, there are different testing versions before the final release and the basic common two are:
  - Alpha version:
    - Often the first “completed” version, but much premature
    - Tested internally
  - Beta version:
    - After Alpha version, approaching to the final version
    - Expected to be tested with targeted end-users

Mobile App Development Process (Evaluation and Distribution Stage)

4) Evaluation and Distribution Stage
   - Evaluate and review a product before distribution and deployment,

- Client's requirements have been met by **providing the testing results**
  - Functional testing
  - Usability testing
  - Performance testing (battery usage, memory, load)
  - Compatibility testing (multiple devices, screen sizes, OS versions)
- The final product is packaged and distributed to the client and targeted end users
- Store listing preparation appropriate documentations and :
  - App icons, screenshots, video previews, User manual, license agreement

Mobile App Development Process (Evaluation and Distribution Stage)

Mobile Application Distributions:

- Android: APK / AAB, signing keys

- Google Play (Google) 

  - An online software store which allows users to browse and download mobile applications

- App Store (iOS/iPadOS, Apple) 
  - Allows users to browse and download apps, developed with Apple's iOS Software Development Kit.

Mobile App Development Process (Evaluation Guide)

For reference

- Towards the end of project, it is important to evaluate the final project output and project development process for future improvement (esp. in student projects)
- Assessing the project *output*, to reflect how well the app has been produced for its original purpose and objectives
  - **Questions**
    - Does the end-user get what is expected, with our solution?
    - Did we successfully integrate the original concept with the actual product elements?
- Assessing the development *process*, to reflect how well the team have gone through the whole process of developing the right product
  - **Questions**
    - Did we have enough exploration of different ideas and concepts? Be creative?
    - Did we make good use of materials and resources available efficiently?

Mobile App Development Process (Operation and Maintenance Stage)

5) Operation and Maintenance Stage
   - After successful deployment and installation, apps (application software) are put in operation
   - Training and support may be included in serious application projects

- **Post-Launch Tasks**
  - Monitoring analytics (crash reports, user behavior, retention)
  - Bug fixes and updates
  - Managing OS upgrades (Android/iOS new versions)
  - Adding new features and enhancements based on user feedback
  - Ensuring app security is up-to-date

- The whole cycle of development process may repeat, if a redesign is needed
  - E.g. for extending the product to a different market

Operational Environment

- Before design and implementation, understanding the operational environment of using a mobile app is important
- Developing a successful mobile application starts with the right context identified first:
  - *What* is the app for?
  - *Who* are the users?
  - *Why* do they use the app, not others?
  - *When*, *where* and *How* are they normally using and interacting?

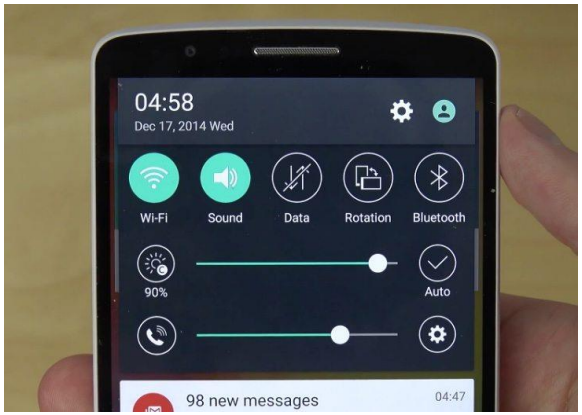
Operational Environment

- Operational environment in the mobile system

1. Mobile Operating Systems
   - Android
   - iOS
2. Device Ecosystem
   - Smartphones, tablets, smartwatches, IoT devices
   - Differences in screen resolution (DPI or pixel density), processing power
3. Network Conditions
   - Mobile data (3G/4G/5G), Wi-Fi, offline mode
   - Handling slow or intermittent connectivity
4. App Sandboxing and Permissions
   - Mobile apps run in isolated environments
   - Permission models: camera, mic, location, notifications, storage

Operational Environment

- Physical environments of operating mobile apps:
  - Unlike using desktop app situations where users are generally operating apps in a static way, people use mobile devices anywhere and situations
  - **Situations with different and changing levels of brightness, noise, temperature, etc.**
    - Outside in the bright sun, inside dark theaters, and even using mobile apps while users are moving from outdoor to indoor with changing environment
    - Attending a lecture or a meeting, taking a rest or shower, jogging or hiking, etc.



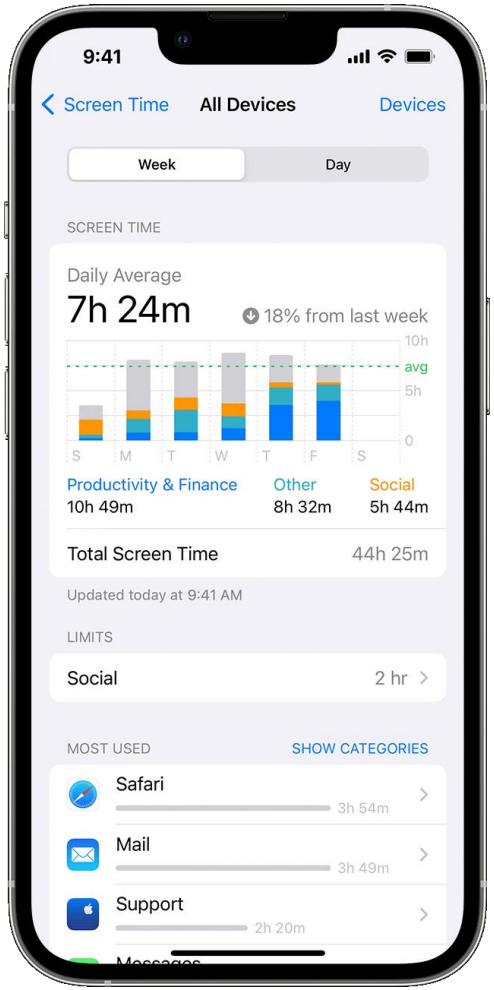
Use under sunlight

Operational Environment

- Social influences:
  - Lots of mobile users will be engaging in conversations or other social activities, while keep a mobile app running
  - People may pass around the device to show people something on-screen or prevent neighborhood seeing your screen due to privacy
  - People may need to suddenly turn off the sound, if it is not socially acceptable to have a noisy device (like in a lecture)

Operational Environment

- Common behaviors of mobile end-users:
  - Limited Time
    - Mobile users won't spend lots of time on their apps in each section of using, most of the time
  - Limited and discontinuous attention:
    - Also lack of attention on their apps in general, but a mobile user may occasionally focus fully on the device like playing a game



Operational Environment

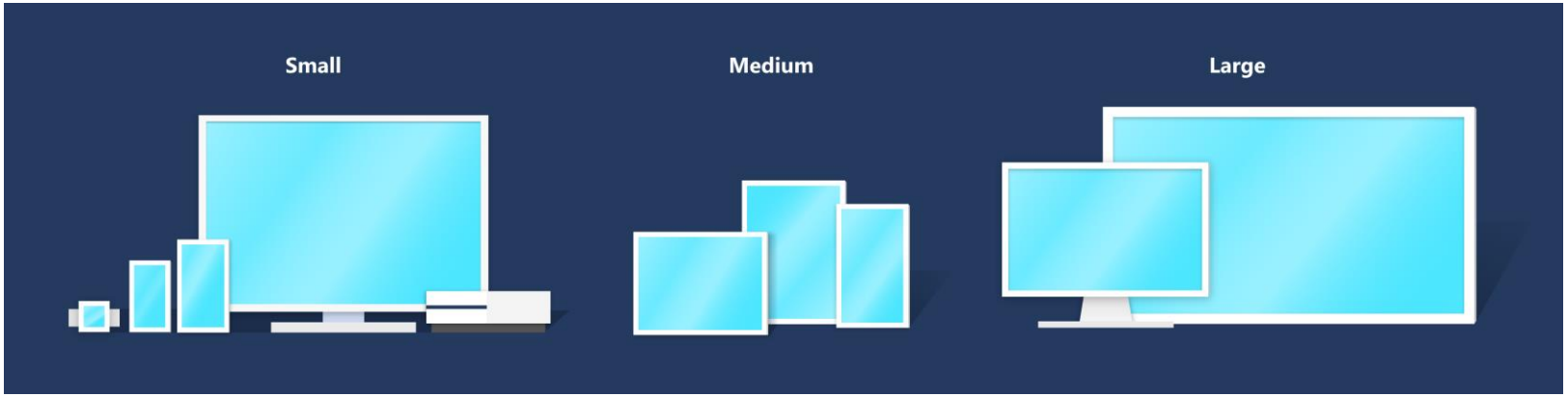
- Common behaviors of mobile end-users:
  - Multi-Tasking
    - Mobile users are often using an app, while doing other task like chatting with friends
    - They may also have more than one app running within the mobile device (if the device supports multi-tasking), e.g. switching between Facebook and playing a game
  - And thus, design for distracted users: make the task sequences easy, quick, re-entrant, and self-explanatory

Hardware and Software Constraints

- Developing apps used in mobile devices face challenges of various hardware and software constraints, depending on the targeted types of devices
- In general, the major hardware constraint is the small size, which causes limited screen display and input area
- Samples of hardware constraints
  - Battery limitations
  - Limited CPU & GPU performance
  - Limited storage
  - Sensor's variability (some devices lack certain sensors)

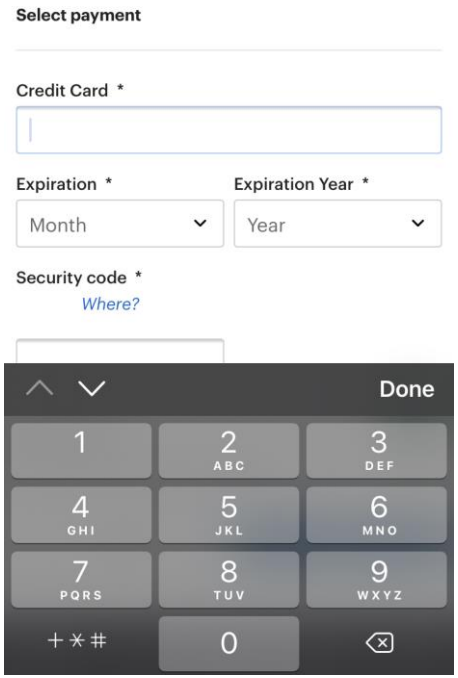
Hardware and Software Constraints

- Tiny screen display sizes:
  - Not much space to present information or choices
  - Restricted content and linearized layout preferred
  - Take away all extra
    - Need to strip the design down to its essence
  - Leave only the most important functions on the front page



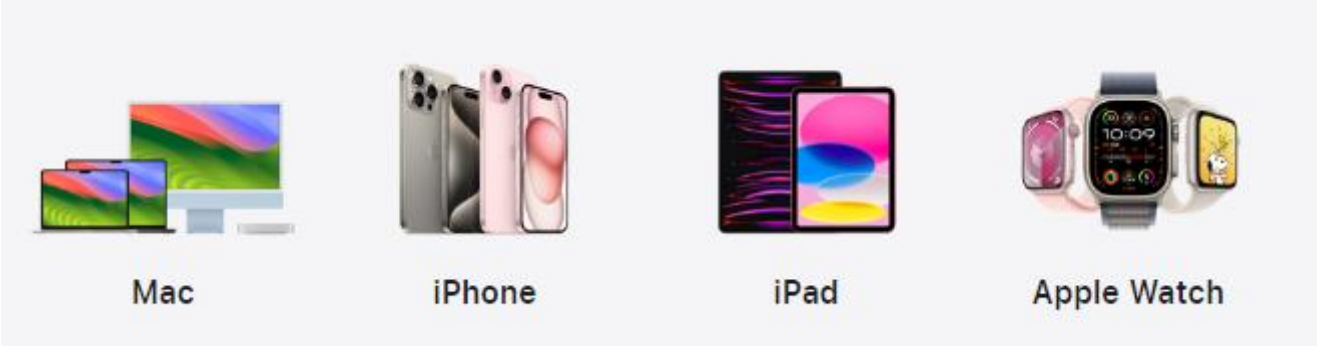
Hardware and Software Constraints

- Small input area:
  - Difficulty of typing text
  - Most access comes from devices with touch screens nowadays
  - As such, there are increasing alternative ways of interacting, e.g. in terms of audio, visual, gesture or movement



Hardware and Software Constraints

- Variable sizes of different devices
  - Non-standardized among devices across different brands, e.g. those with Android OS
  - Developing universal apps for supporting them requires extra efforts
- Limited hardware capacities
  - Further consideration is needed about the comparatively limited hardware capacities, e.g. Memory and CPU speed
  - Do not simply create a “mini”-version of desktop application



Hardware and Software Constraints

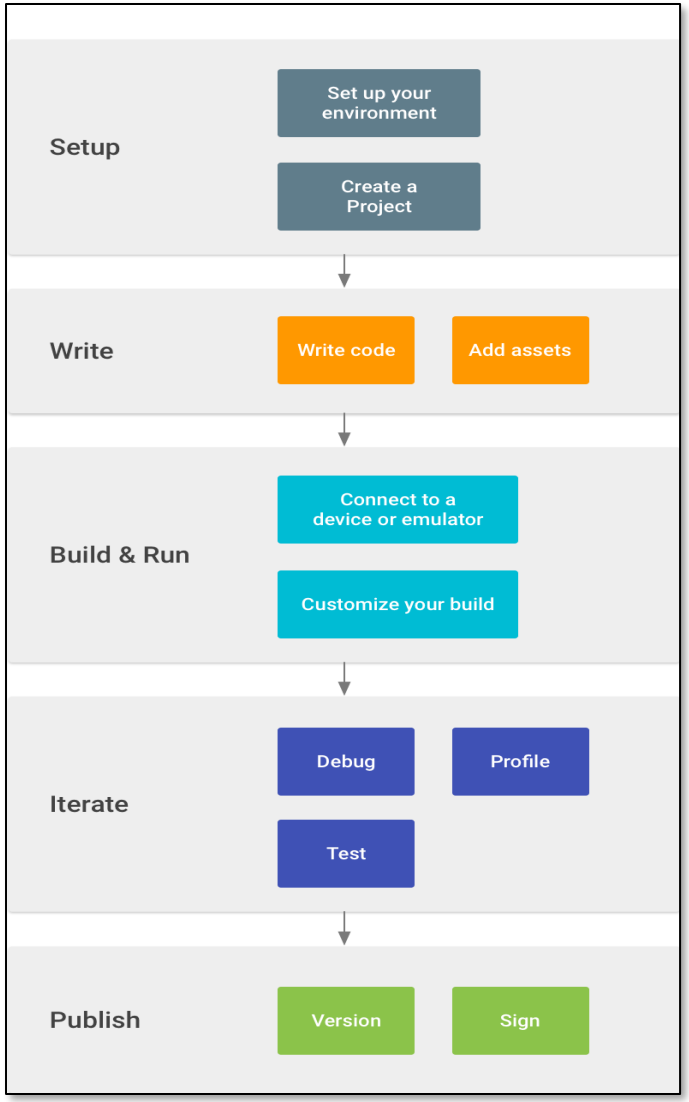
- Software limitations correspondingly further complicate the development considerations
  - Because of limited hardware capacities, mobile software are supported with fewer or smaller software frameworks / APIs.
- Application programming interface (API)
  - is a set of defined rules and interfacing settings that enable different applications to communicate with each other. It acts as an intermediary layer that processes data transfers between systems. (from IBM)
- **Samples of software constraints**
  - RAM management and background execution limits
  - App lifecycle restrictions (e.g., Android activity lifecycle)
  - API fragmentation (especially Android)
  - Platform policies and security restrictions

Hardware and Software Constraints

- **Implications for Developers**
  - Optimize performance and memory usage
  - Minimize network requests
  - Use efficient image compression
  - Handle app lifecycle gracefully

Developing Android App (Typical Workflow of Developing App)

- A typical implementation (development) of mobile apps includes 5 phases:
  1) Setup the workspace
  2) Write the app
  3) Build and run
  4) Iterate (Debug & test)
  5) Publish



Developing Android App (Typical Workflow of Developing App)

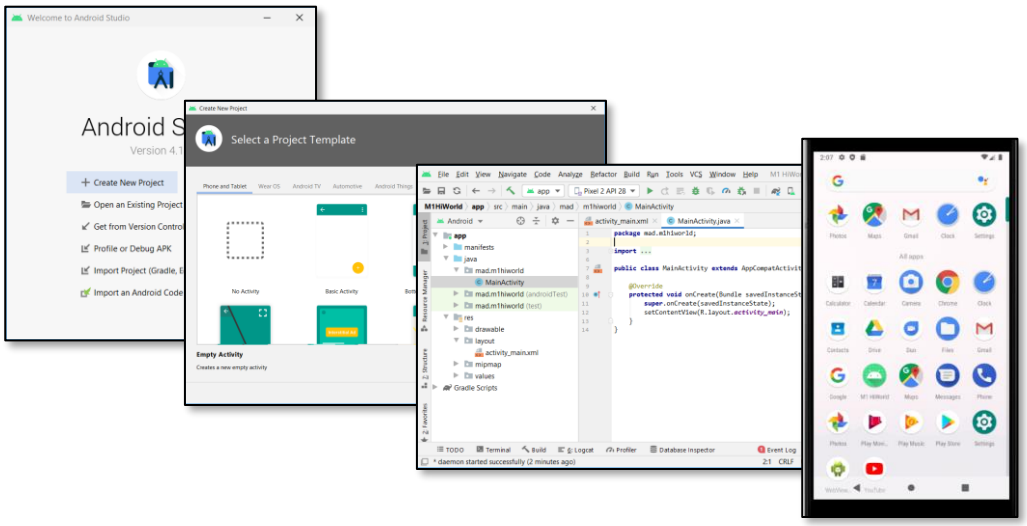
- **Setup Phase**
  - Install Android Studio
  - Configure SDKs, emulators, debugging tools
- **Project Structure of App**
  - **Manifest.xml**: app permissions, components
  - **res:** layouts, drawables, strings
  - **Java** or kotlin: activity and logic classes
  - Gradle build configuration
- **UI Development**
  - XML layouts or Jetpack Compose
  - Material Design components
  - Supporting multiple screen densities

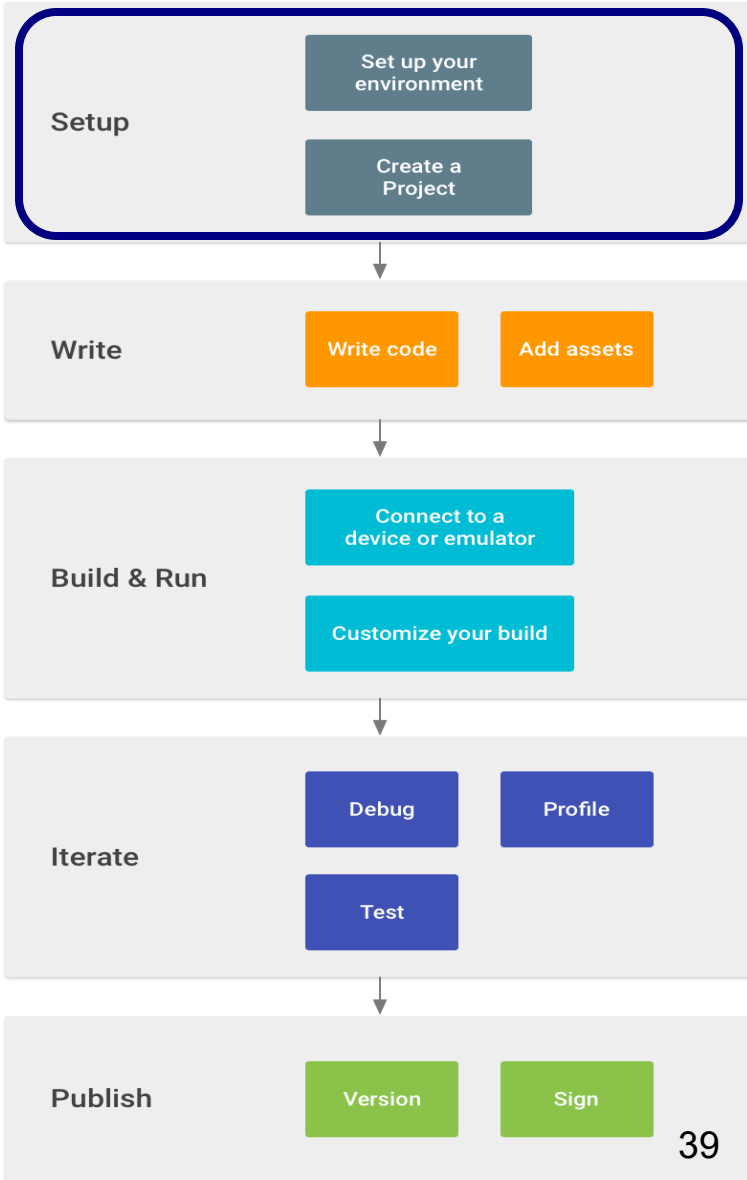
(Typical Workflow of Developing App)

- **Core Development**
  - Activities, Fragments, ViewModels
  - API
  - Local storage using Room database
- **Testing & Debugging**
  - Unit testing
  - UI testing
  - Debugging
- **Packaging and Deployment**
  - Generate signed AAB/APK
  - Upload to Google Play Console
  - Configure release notes, device targeting, closed testing groups

Typical Workflow of Developing App (Android)

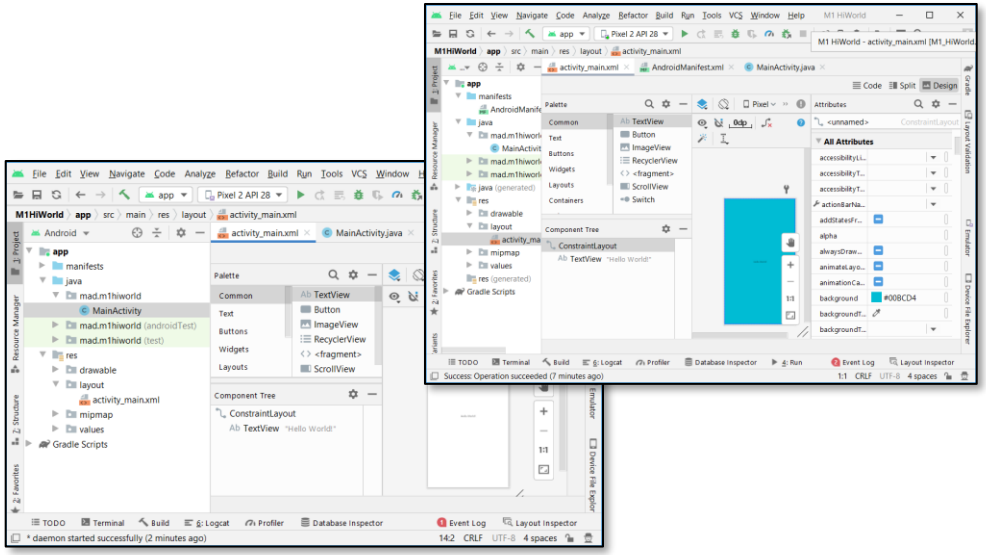
- Setup the workspace (development environment) for developing app, e.g.
  - Install Android Studio
    - Latest Version Otter | 2025.2.1
  - Create a project
  - Create Android Virtual Devices (AVDs) for testing, e.g. using AVD `Pixel 3` with `API 24`
  - *(* Android Studio often updates new version for debug or improvement.)*

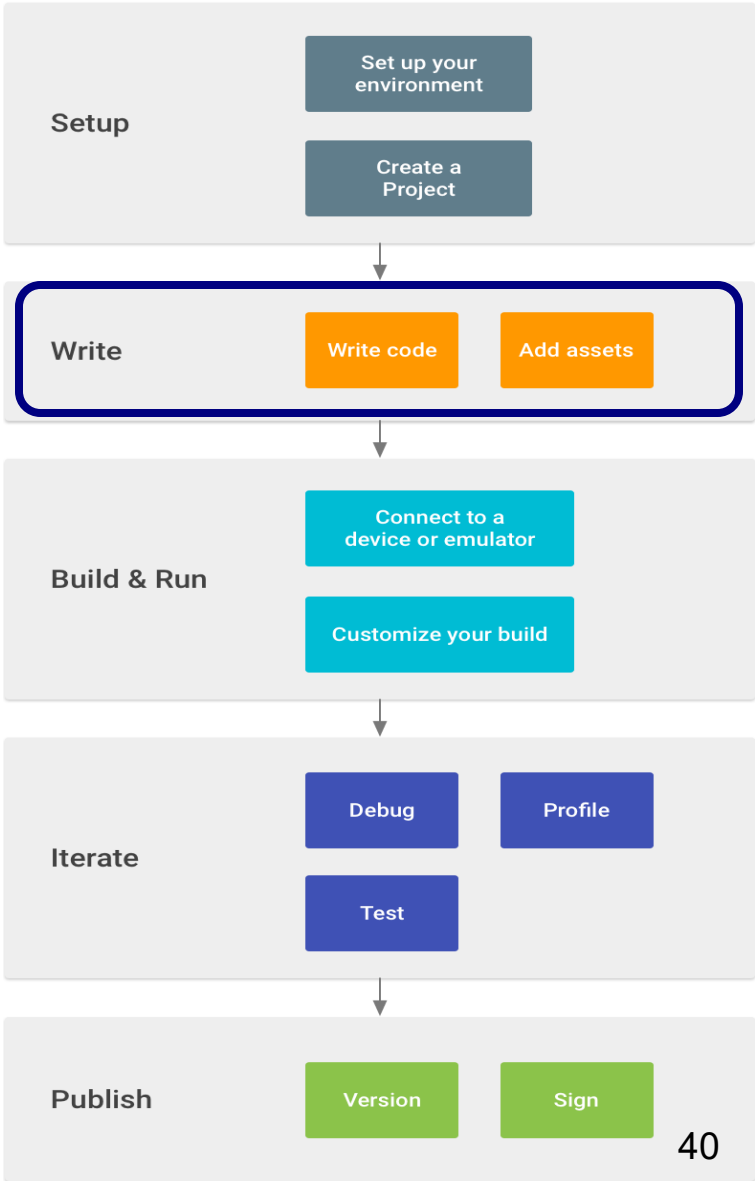




Typical Workflow of Developing App (Android)

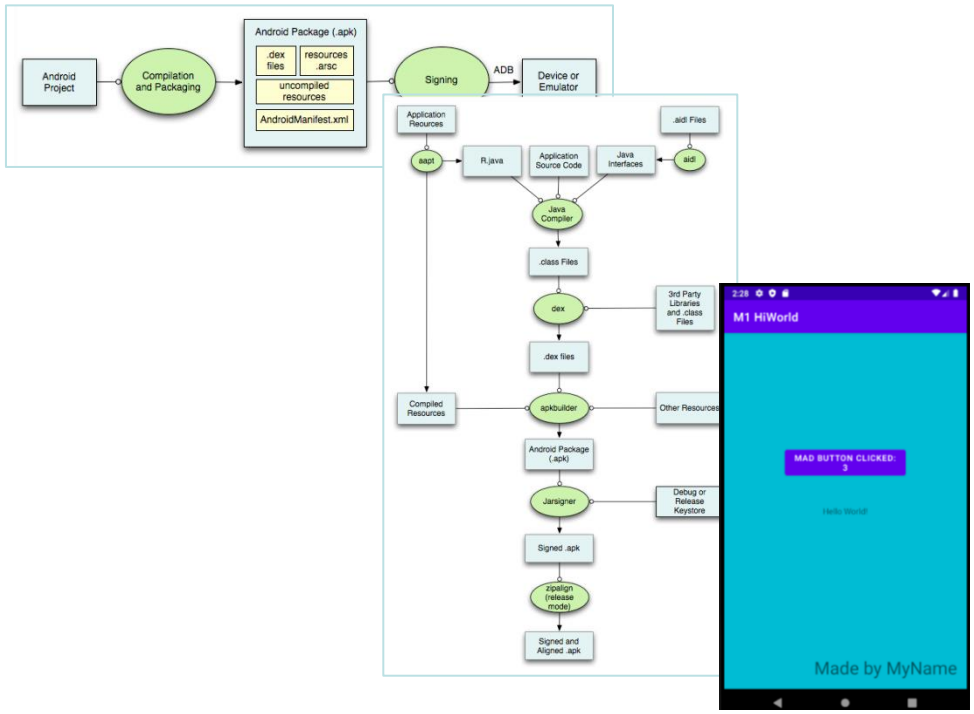
- Write the app (Development & Implementation), e.g.
  - Write code
  - Design UI/UX
  - Create resources for different device types

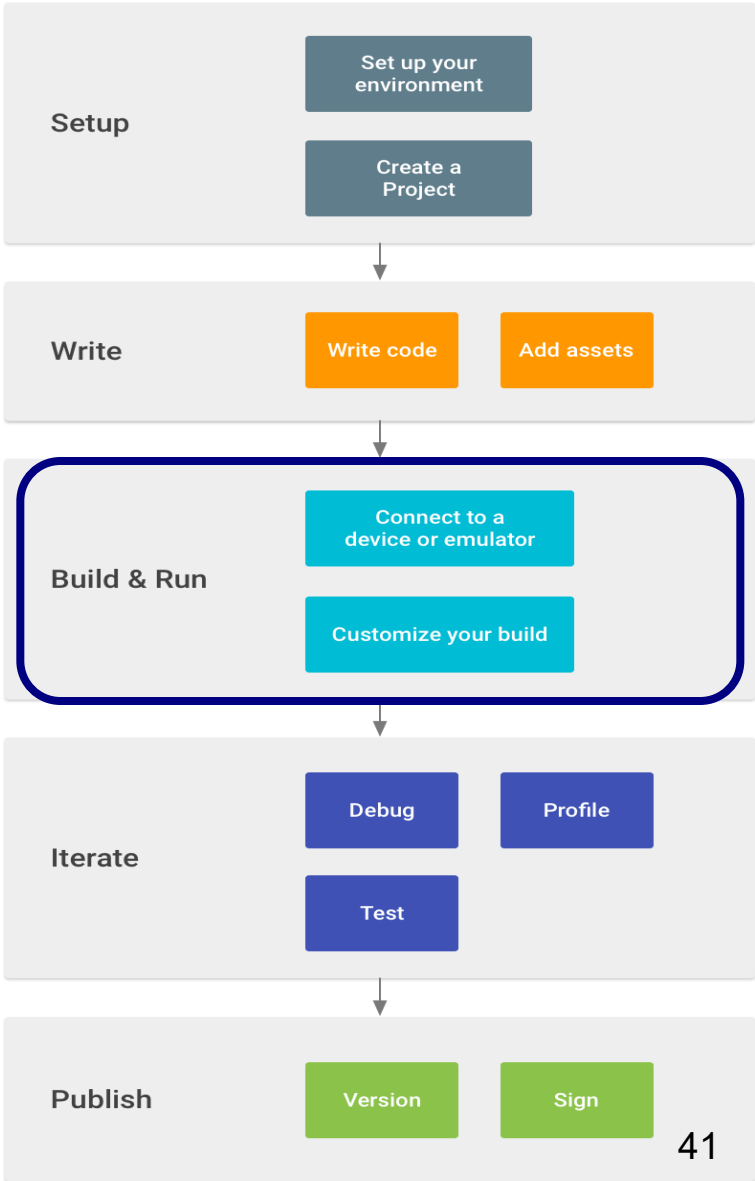




Typical Workflow of Developing App (Android)

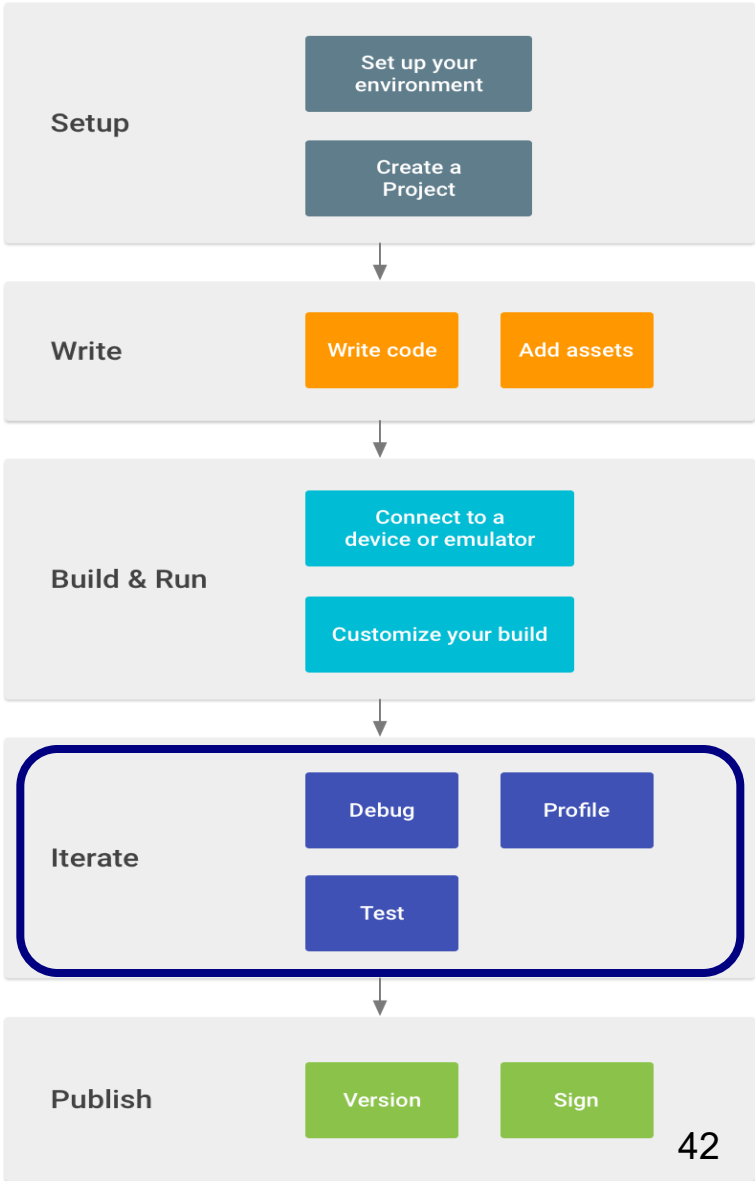
- Build and run
  - Build the project into a debuggable `.apk` package
  - Install and run on the emulator or an Android-powered device, e.g. using AVD `Pixel 3` with `API 24`





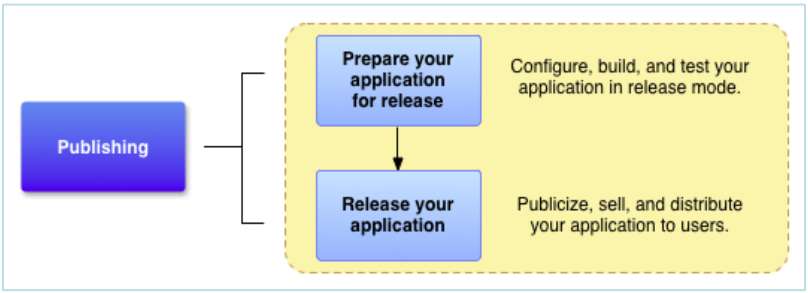
Typical Workflow of Developing App (Android)

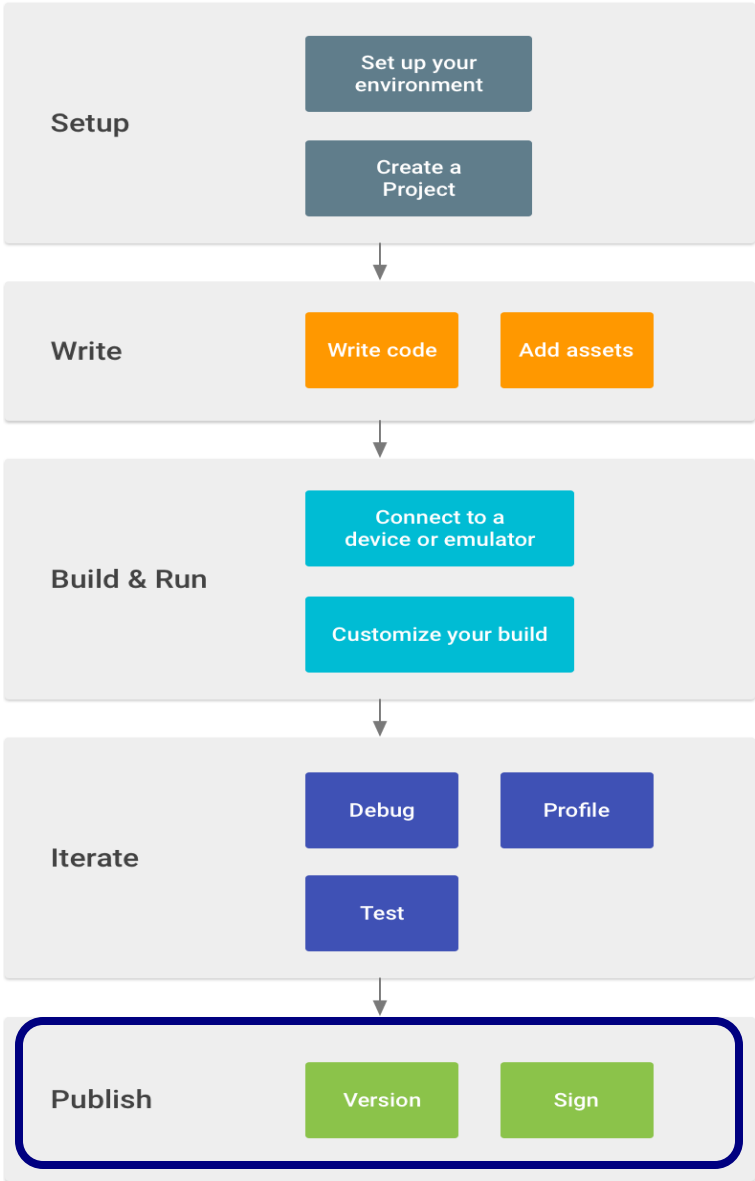
- Iterate (Debug & test)
  - Continue writing (debugging) the app, with a focus on:
  - Eliminating bugs, and
  - Optimizing app performance



Typical Workflow of Developing App (Android)

- Publish
  - Configure and build the application for release and distribute the application to end users
  - The release-version should be signed with developer's certificate





References

- Part of this slide set is prepared and extracted from the followings:
  - Fling, B. (2009) *'Mobile Design and Development: Practical techniques for creating mobile sites and web apps'*, O'Reilly Media, Inc.
  - Tidwell, J. (2011) *'Designing Interfaces'* (2ed), O'Reilly Media, Inc.
  - Google Inc. 'Website for Android Developers', http://developer.android.com/
  - Google Inc. 'Website for Android Sources', https://source.android.com/
  - Apple Inc. 'Website for Apple Developers', Website: https://developer.apple.com/
  - Microsoft Corp. 'Website for Windows Developers', https://developer.microsoft.com/en-us/windows
  - Wikipedia Website: http://en.wikipedia.org
- This set of slides is for teaching purpose only