

Mobile Application Development

Produced
by

David Drohan (ddrohan@wit.ie)

Department of Computing & Mathematics
Waterford Institute of Technology

<http://www.wit.ie>

<http://www.jumpyjosh.com>



Waterford Institute of Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE



Mobile App Design



iOS



Android



Windows



Agenda

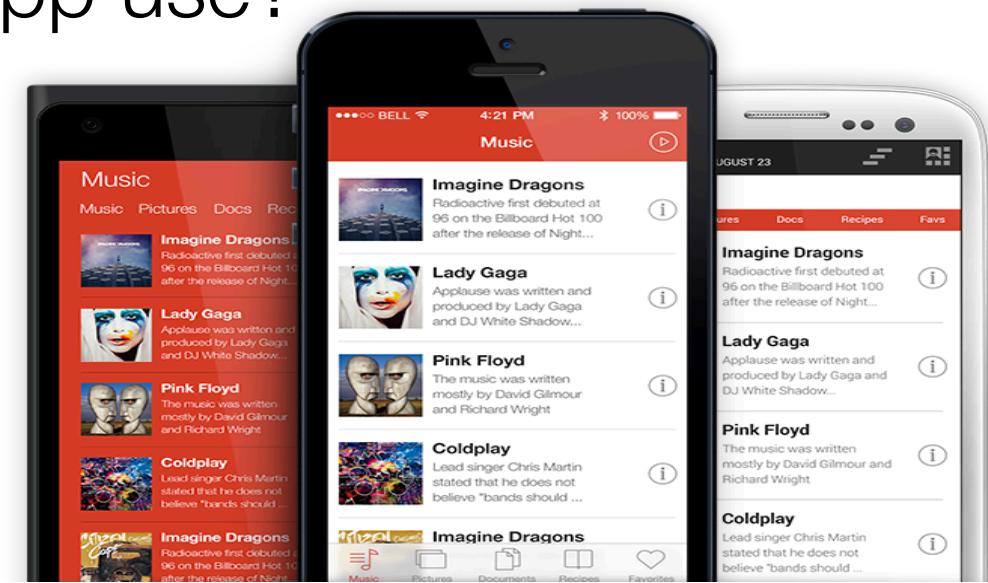
- ❑ App Design Overview
- ❑ Know your Framework
- ❑ Choose a Methodology
- ❑ Initial Design & Action Plan
- ❑ Design the User Interface





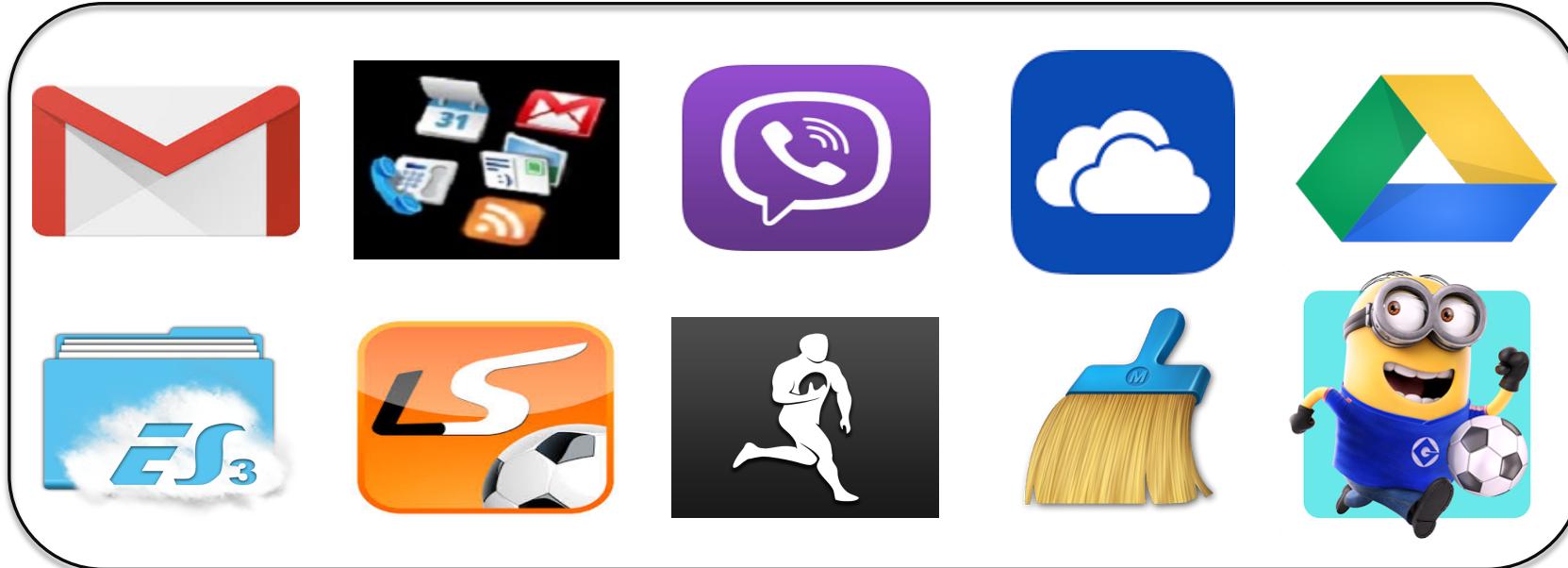
Overview

- ❑ Before writing any code, you must ask yourself some fundamental questions:
 - What is the App's purpose?
 - What are the App's features?
 - What kind of data model will the App use?
 - Is it a universal App?



Overview

Generally, an app that is well designed, has features that users find both appealing, appropriate, and useful.



Overview - Design Your App with Care

- ❑ App Design does not *just* mean user interface design
- ❑ App Design does not *just* mean code
- ❑ It begins with some high-level information and ideas
- ❑ You then think about how to best implement these ideas





App Design Frameworks

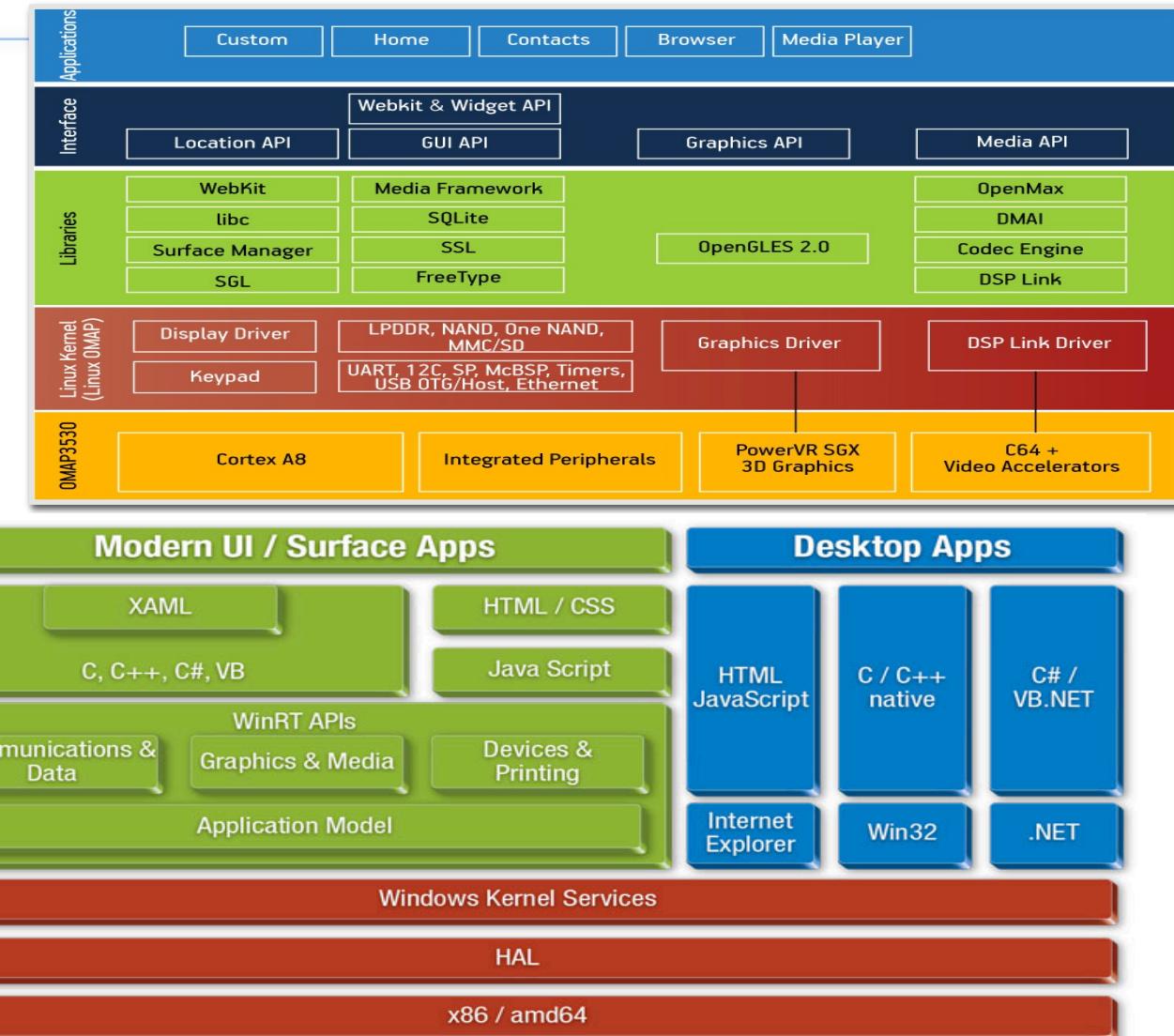
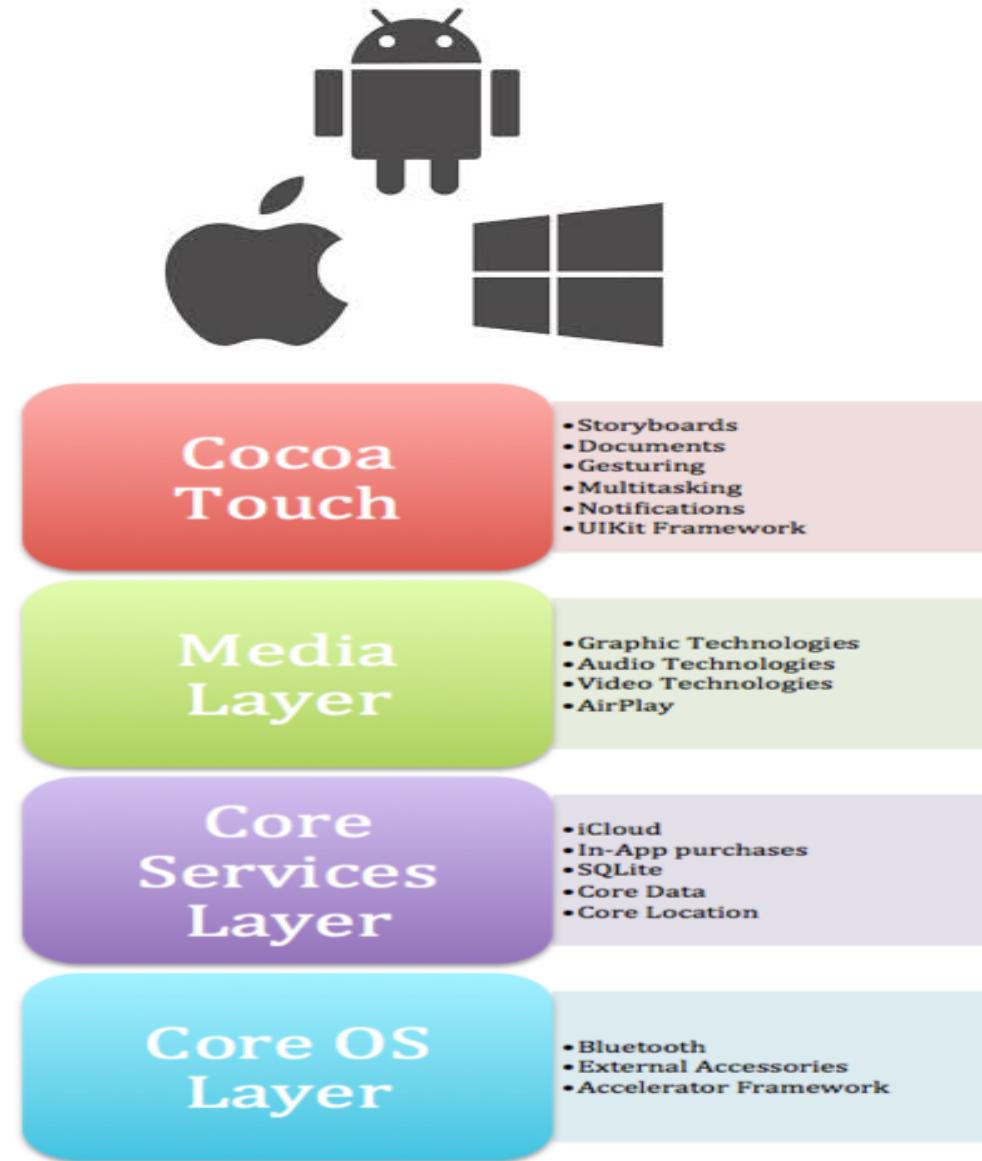
You also need to become familiar with the respective frameworks.

- Your app depends on the objects you can create with these frameworks.
- Get to know the framework objects that implement the basic structure of an app, that serve as the building blocks of your data model, and that compose the unique experience your app presents to users.





App Design Frameworks





App Design Methodology

❑ We'll try and follow SCRUM

- Scrum is a management framework for incremental product development using one or more cross-functional, self-organizing teams of about seven people each.
- It provides a structure of roles, meetings, rules, and artifacts. Teams are responsible for creating and adapting their processes within this framework.
- Scrum uses fixed-length iterations, called Sprints, which are typically 1-2 weeks long (never more than 30 days). Scrum teams attempt to build a potentially shippable (properly tested) product increment every iteration.

❑ An Alternative to Waterfall

- Scrum's incremental, iterative approach trades the traditional phases of "waterfall" development for the ability to develop a subset of high-value features first, incorporating feedback sooner.

scrumreferencecard.com

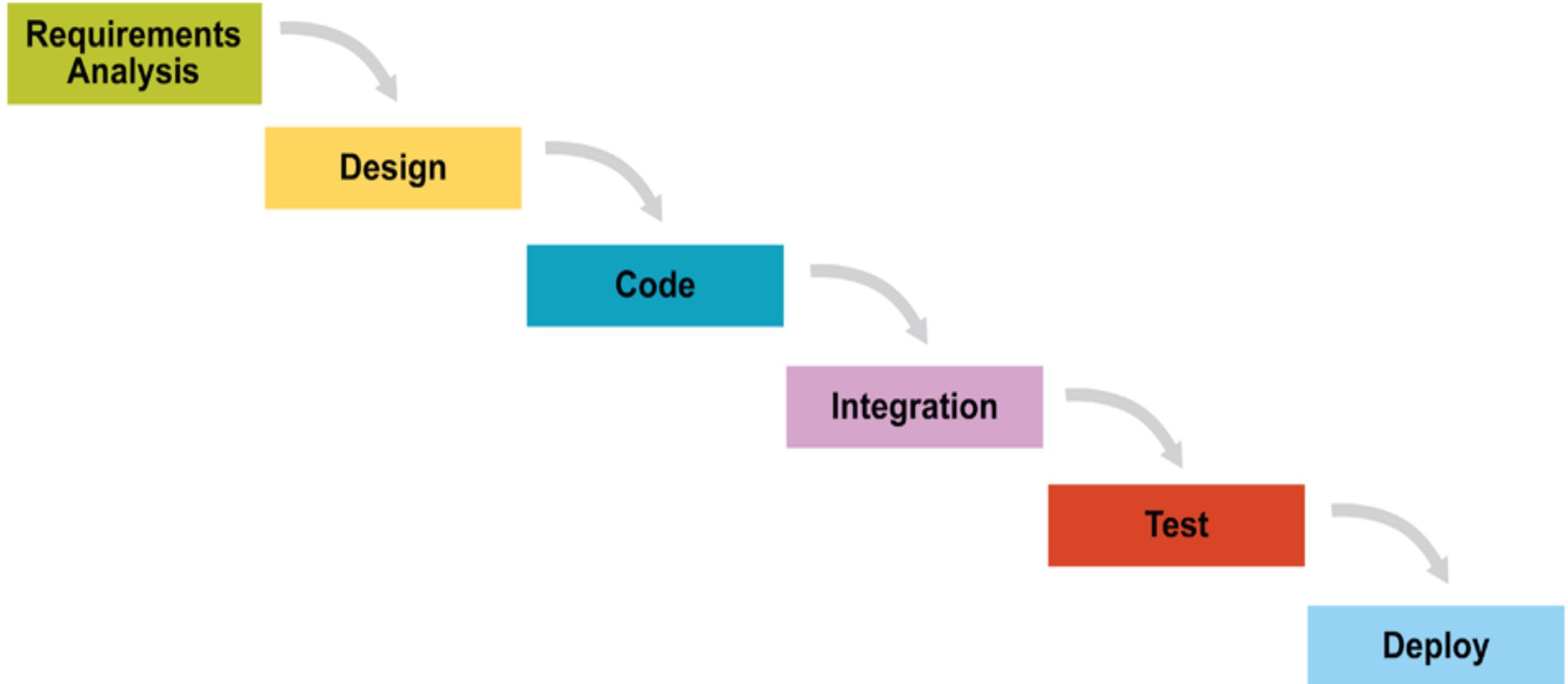


Figure 1: Traditional “Waterfall” development depends on a perfect understanding of the product requirements at the outset and minimal errors executing each phase.

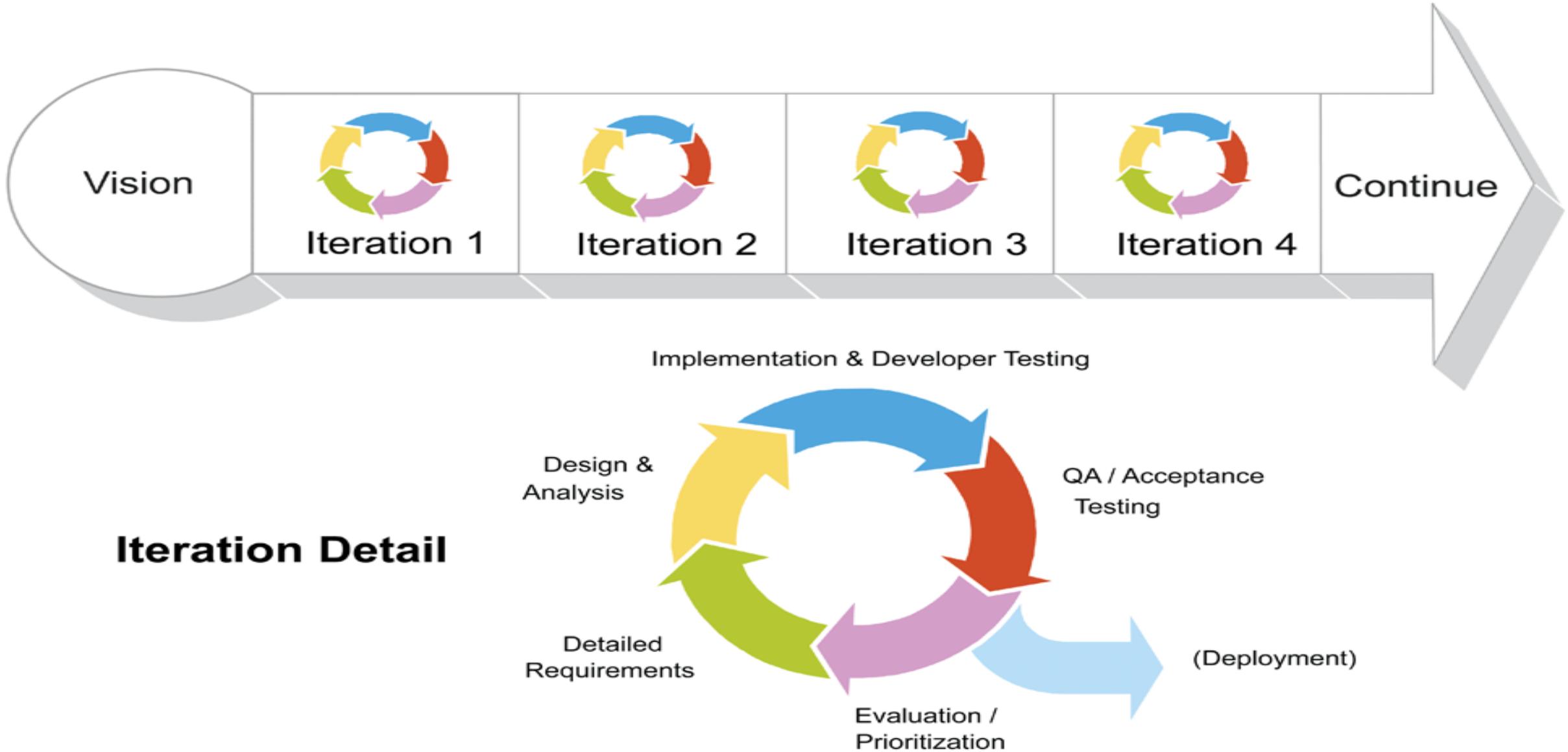


Figure 2: Scrum blends all development activities into each iteration, adapting to discovered realities at fixed intervals.



CoffeeMate | Trello

The screenshot shows a Trello board titled "CoffeeMate". The board has three main columns:

- Add a Coffee**: Contains cards for "Design Add Screen", "Code Screen Transition", "Code Coffee Data Capture", and "Code Coffee Persistence".
- List All Coffees**: Contains cards for "Design List Screen", "Code Screen Transition", and "etc...".
- Update a Coffee**: Contains cards for "Design Update Screen", "Code Screen Transition", and "etc...".

A sidebar on the right is open, showing the following information:

- Members**: Shows a user "DD" and a button "Add Members...".
- Activity**: Lists recent actions:
 - David Drohan added etc... to Update a Coffee. Nov 20 at 11:08
 - David Drohan added etc... to List All Coffees. Nov 20 at 11:08
 - David Drohan made this board visible to the public. Nov 20 at 11:06
 - David Drohan added Code Screen Transition to List All Coffees. Nov 20 at 11:06
 - David Drohan added Design List Screen to List All Coffees. Nov 20 at 11:06

Figure 3: CoffeeMate | Trello – example of using a Scrum tool to detail the iterations.



Where to Start? Do Your Initial Design

- ❑ A good app starts with an idea that is expanded to more fully-featured product description
- ❑ Early in the design phase, it helps to understand just WHAT you want your app to do (not how you will do it)
- ❑ Write down the set of high-level features that would be required to implement your idea
- ❑ Prioritise those features based on what you think your users will need
- ❑ Do a little research into the framework (iOS/Android etc.) itself so that you understand its capabilities and how you might be able to use them to achieve your goals



Your Initial Design

- Sketch out some rough interface designs on paper to visualise how your app might look
- A pen and paper may be the most advanced tech you need at this point
- There are lots of app design tools out there to assist you.

www.smashingapps.com

The screenshot shows the homepage of WireframeSketcher. At the top, there's a navigation bar with links for Tour, Download, Buy, Blog, Gallery, Support, and About. Below the navigation is a blue header bar with the text "Sample Mockups". Underneath is a yellow navigation bar with Sample Mockups, Features, and Testimonials. The main content area features the heading "Here's what you can create with WireframeSketcher" followed by eight sample mockups arranged in a grid:

- Android Application: Shows a phone screen with icons for Gmail and Calendar.
- Facebook Website: Shows a browser window displaying a Facebook profile for "John Doe".
- iPhone App: Shows a phone screen with a back button and a "facebook" button.
- iTunes: Shows a screen with a play button and a library section for Music and Movies.
- iPad Mockup: Shows a tablet screen with a back button and a search bar.
- Windows Phone Mockup: Shows a phone screen with a "CHOOSE A C" button.
- LinkedIn Website: Shows a browser window displaying a LinkedIn profile.
- Skype - Mr Anderson: Shows a Skype interface with contact lists and search functions.



Your Initial Design

- ❑ The goal of your initial design is to answer some very important questions about your app.
- ❑ The set of features and the rough design of your interface help you think about what will be required later when you start writing code.
- ❑ At some point, you need to translate the information displayed by your app into a set of data objects.
- ❑ Similarly, the look of your app has an overwhelming influence on the choices you must make when implementing your user interface code.
- ❑ Doing your initial design on paper (rather than on the computer) gives you the freedom to come up with answers that are not limited by what is easy to do.

Next Step? Create an Action Plan

- ❑ iOS & Android assumes that all apps are built using the **Model-View-Controller** design pattern. Therefore, the first steps you can take toward achieving this goal are to choose approaches for the data and view portions of your app.
-
1. Choose a basic approach for your data model
 2. Decide whether you need support from Framework specific elements
 3. Choose an approach for your user interface





Create an Action Plan

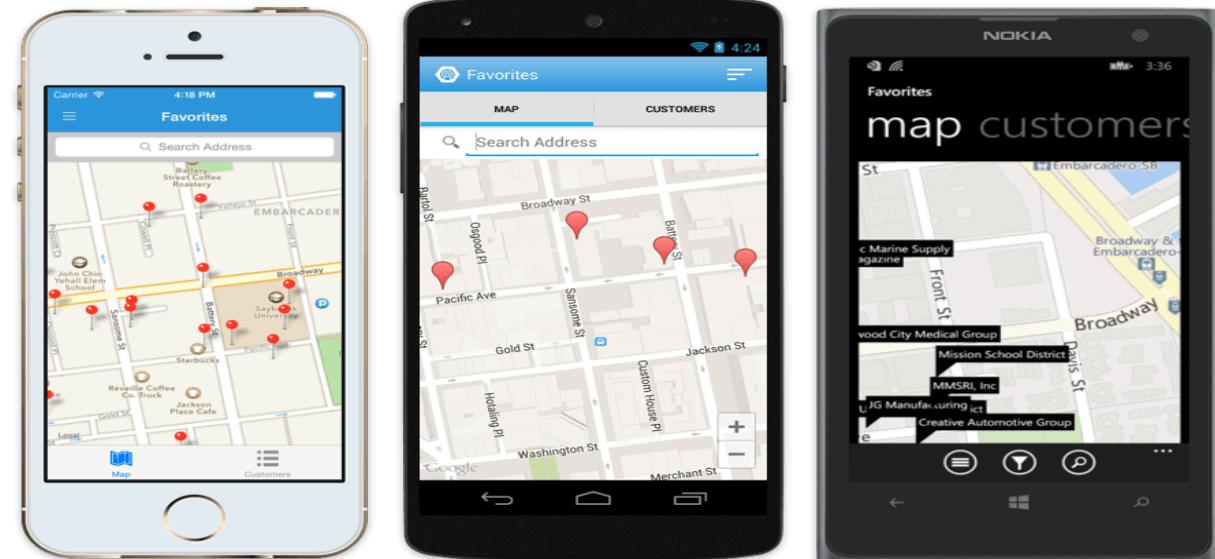
1. Choose a basic approach for your data model

- ❑ **Existing data model code**—If you already have data model code written in a C-based/java based language, chances are you can integrate that code directly into your apps.
- ❑ **Custom objects data model**—A custom object typically combines some simple data (strings, numbers, dates, URLs, and so on) with the business logic needed to manage that data and ensure its consistency.
- ❑ **Structured data model**—If your data is highly structured—that is, it lends itself to storage in a database—use Core Data (iOS) or SQLite (iOS/Android) to store the data. Core Data provides built-in support for some advanced features like undo and iCloud. (SQLite files cannot be used in conjunction with iCloud.)



Create an Action Plan

2. Decide whether you need support from Framework specific elements
 - ❑ Each Framework will have a set of elements/components which you are free to use and integrate into your own app, or even customize to do a specific job.
 - ❑ Get to know how these components work and how they can make your app more useful and user-friendly.





Create an Action Plan

3. Choose an approach for your user interface

- ❑ **Building block approach**—The easiest way to create your user interface is to assemble it using existing **view objects**. Views represent visual elements such as tables, buttons, text fields, and so on. You use many views as-is but you can also customise the appearance and behavior of standard views as needed to meet your needs. You can also implement new visual elements using custom views and mix those views freely with the standard views in your interface. The advantages of views are that they provide a consistent user experience and they allow you to define complex interfaces quickly and with relatively little code.
- ❑ **OpenGL ES-based approach**—If your app requires frequent screen updates or sophisticated rendering, you probably need to draw that content directly using OpenGL ES. The main use of OpenGL ES is for games and apps that rely heavily on sophisticated graphics and therefore need the best performance possible

UI Builders



The collage illustrates four different approaches to mobile application user interface (UI) design:

- Top Left (Xcode):** Shows the Xcode interface with the Project Navigator on the left, the PreferencesController.xib storyboard in the center, and the PreferencesController.h header file in the Editor tab on the right.
- Top Right (Blend for Visual Studio 2013):** Shows Blend for Visual Studio 2013 with the PhoneApp1.sln solution open. It displays the MainPage.xaml XAML code, the visual designer showing a WP8 application page titled "Phone App", and the Properties panel.
- Bottom Left (Eclipse ADT):** Shows the Eclipse IDE with the Package Explorer on the left, the Java code for HelloActivity.java in the center, and the graphical layout editor on the right showing a smartphone screen with the "Hello world" message.
- Bottom Right (Android Studio):** Shows the Android Studio interface with the Java code for activity_hello.xml in the center, the graphical layout editor on the right showing a smartphone screen with the "Hello world" message, and the LogCat and Device Manager tabs at the bottom.



What's Next? Start the App Creation Process

- When starting to actually code your app, you should have answers to the following questions in mind:
 - **What is the basic interface style of your app?** Different types of app require different sets of initial views and view controllers. Knowing how you plan to organise your user interface lets you select an initial project template that is most suited to your needs. You can always change your user interface later, but choosing the most appropriate template first makes starting your project much easier.
 - **Do you want to create a universal or targeted app?** Creating a universal app requires specifying different sets of views and view controllers for iPad and iPhone and dynamically selecting the appropriate set at runtime. Universal apps are nice because they support more iOS devices but do require you to factor your code better for each platform.
 - **Do you want your app to use storyboards?** Storyboards simplify the design process by showing both the views and view controllers of your user interface and the transitions between them. Storyboards are supported in iOS 5 and later and are also a feature within Android Studio.
 - **What do you want to use for your data model?** Some types of apps lend themselves naturally to a structured data model, which makes them ideal candidates for using Core Data or SQLite.



Start the App Creation Process

❑ The following phases of app development are quite common:

- Start writing your app's primary code
- Add support for app state changes
- Create/include the resources needed to support your app 
- As needed, implement any app-specific behaviors that are relevant
- Add the advanced features that make your app unique
- Do some basic performance tuning for your app



What about The User Interface?



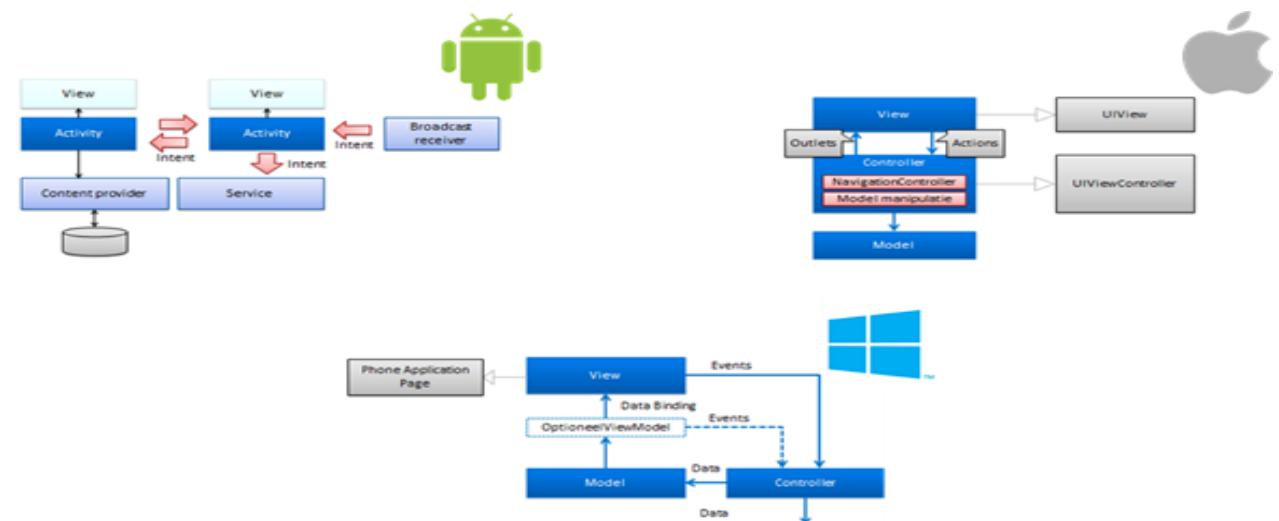
The User Interface

- ❑ Your app's user interface is everything that the user can see and interact with.
- ❑ Following your platform's **UI Design Guidelines** is a tried-and-trusted way to increase metrics like user retention and customer satisfaction. It makes it easier for your users to learn how to use the application to the fullest extent, as quickly and intuitively as possible.
- ❑ Be mindful of the fact that the respective UI Frameworks may use different 'widgets' to do the same/similar things, and position user options differently.



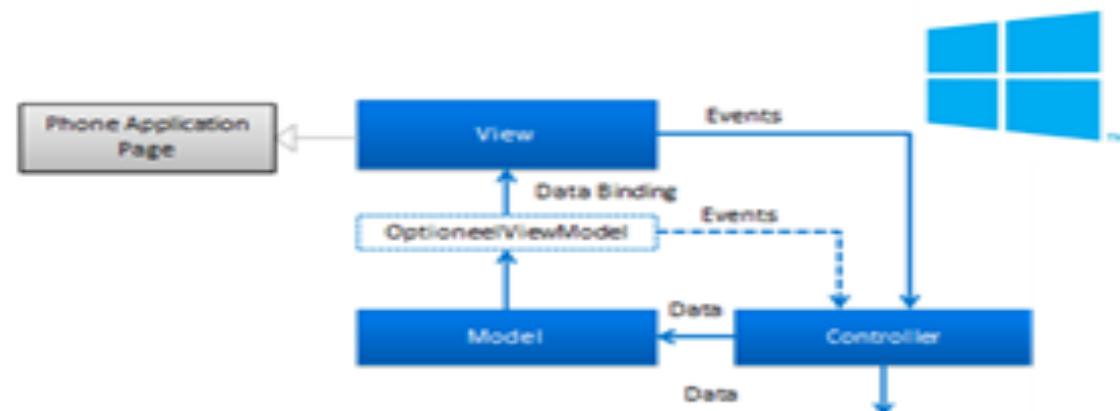
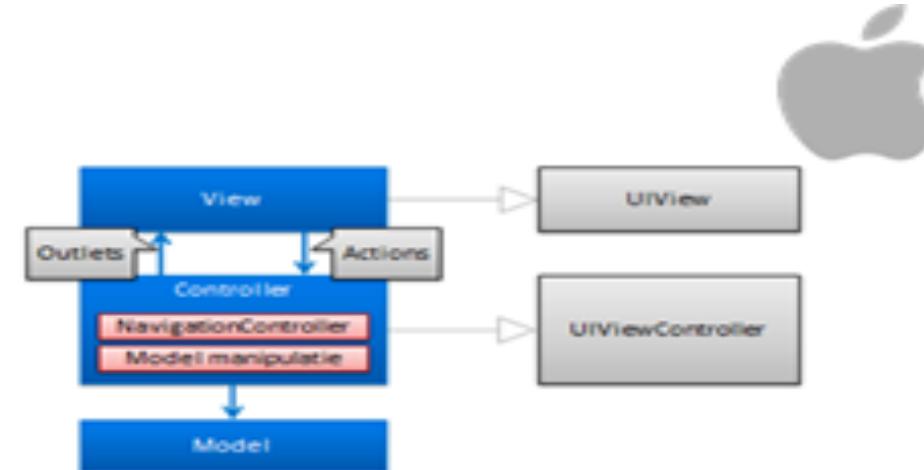
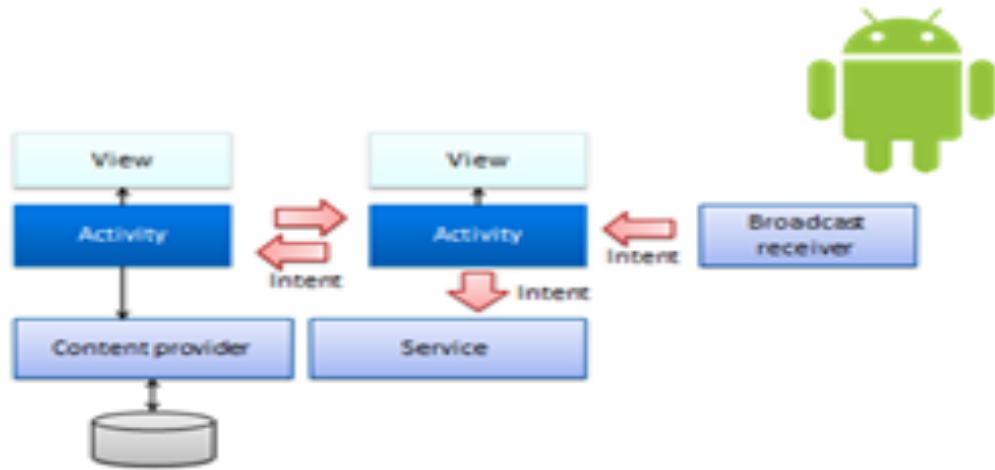
The User Interface

- Make sure you are familiar with the relevant **UI Design Guidelines** so the ‘look-and-feel’ of your app is consistent with the target platform.
- Also, you need to know the respective UI Frameworks and hierarchies, and how they assist you when building your User Interface.

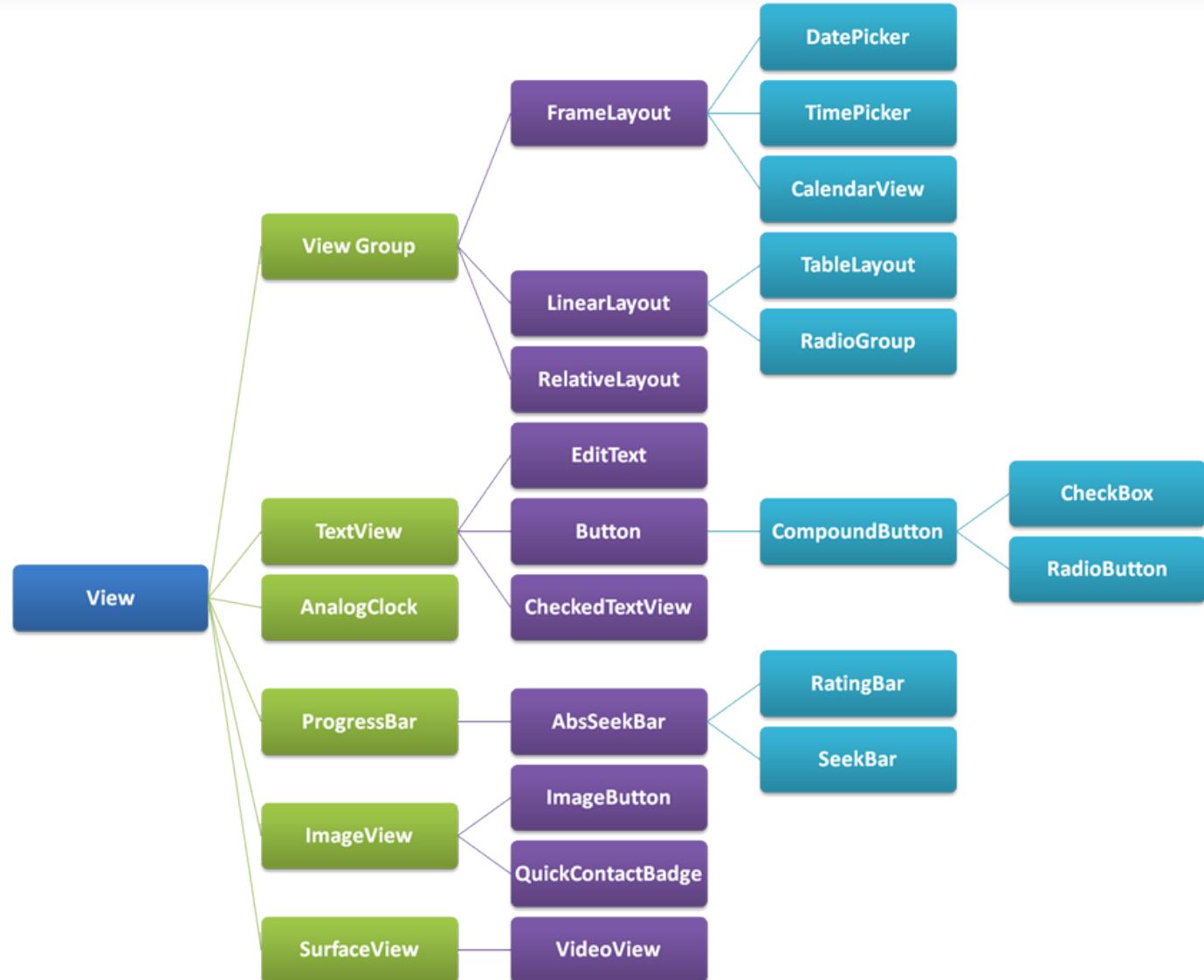




The User Interface



Android *View* UI Hierarchy



Summary

- ❑ App Design Overview
- ❑ Know your Framework
- ❑ Choose a Methodology
- ❑ Initial Design & Action Plan
- ❑ Design the User Interface





More Information - iOS

- ❑ [iOS Technology Overview](#) describes the frameworks and other technologies that are available to your app in iOS.
- ❑ [iOS Human Interface Guidelines](#) teaches you how to make your app consistent with the user-interface conventions for iOS.
- ❑ [Developing for the App Store](#) walks you through the process of developing apps, provisioning devices for testing, and submitting apps to the App Store.
- ❑ [Programming with Objective-C](#) describes how to define classes, send messages, encapsulate data, and accomplish a variety of other tasks with the Objective-C programming language.
- ❑ [iOS App Programming Guide](#) explains the essential things you must know and do when developing an iOS app.



More Information - Android

- ❑ [Android Developer Site](#) is the official site and provides the SDK, Developer's Guide, Reference, and Android Market for the open source project.
- ❑ [Android Design Guidelines](#) teaches you how to make your app consistent with the user-interface conventions for Android.
- ❑ [Android Programming Guide](#) will give you everything you need to get started with your first Android app.
- ❑ [Google Play Developer Console](#) enables developers to easily publish and distribute their applications directly to users of Android-compatible devices.



More Information – Windows 10

- [Windows 10 Developer Center](#)



Questions?