

# UNIT – I

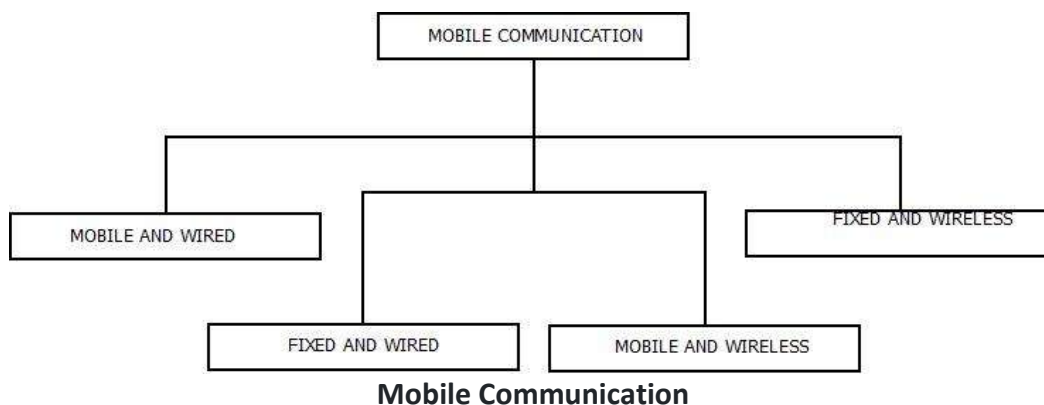
## Introduction to Mobile Computing

Mobile Computing is a technology that allows us to transmit data, audio, and video via devices that are not connected with any physical link. The key features of mobile computing are that the computing devices are portable and connected over a network.

Mobile computing began with the first laptops developed in 1980, and from there, rapidly grew in capability with the 640\*640 portable laptops from Apple in 1990, the development of the first PDA in 1993, the first Smartphone from IBM released in 1994, network connectivity enabled smartphone in 2000, the first iPhone released in 2007, and the first android smartphone in 2009.

Today there is great variety among mobile computing devices, and their capabilities are increasing with each new model released. The major elements of mobile computing are mobile communication, mobile hardware, and mobile software.

- **Mobile Communication:** This refers to the communication infrastructure set in place, including wireless network infrastructure, protocols, data formats, bandwidths, and portals necessary to ensure seamless connectivity and communication.
  - Mobile communication framework includes communication device such as mobiles, laptops, as rules of conduct, fitness etc. They are responsible for delivering of smooth communication process.
  - Mobile communication can be of one of the following form as mentioned below.



1. **Mobile and Wired :** In this configuration, Some of the devices are wired and some are mobile in nature. For example : Laptops.

2. **Fixed and Wired** : In this configuration, The devices are fixed at a position and are connected through a physical link for communication. For example : Office/Desktop Computer.
  3. **Mobile and Wireless** : In this configuration, devices can communicate(data transmission) with each other irrespective of their position and can connect to any network without the use of any wired device. For example : WiFi Dongle.
  4. **Fixed and Wireless** : Fixed wireless is the operation of wireless communication devices or systems used to connect two fixed locations (e.g., building to building or tower to building) with a radio or other wireless link, such as laser bridge. Usually, fixed wireless is part of a wireless LAN infrastructure.
- **Mobile Hardware:** The hardware is the mobile computing devices and supporting devices, with the capabilities required to perform their required operations and connect to networks.
  - **Mobile Software:** The most important software component is the operating system, which is the brain of any computing system. For a laptop, this may be Windows, Linux or macOS, and for a smartphone, it may be Android or iOS. The different applications running on a device are also part of the mobile software.

## **Mobile Computing Devices**

Usually, a mobile computing device would have a body- made of metal or plastic, a RAM, a CPU, a hard drive, a motherboard, a keyboard and a mouse- which could be separate components in the body or touch-based, a screen, a video card, an operating system, software applications, and finally, a network connection.

This is around the same as the components of a personal computer, which isn't a mobile device. But mobile devices may have other components too, to make them portable , and certain characteristics that make them different-

- **Size:** The portability of mobile devices demands a smaller size. Reduction in size without reducing capabilities has also always been a challenge when developing mobile devices.
- **Power Source:** Mobile devices are usually powered by rechargeable batteries. Improving the battery life of mobile devices is another significant area of research.
- **Operating System:** Laptops run on more or less the same OS as PCs, but for smartphones and other devices, the OS is significantly different. They are powerful but scaled-down and made specifically for particular devices.
- **Connectivity:** Mobile computing devices have capabilities that allow access to the internet. Also, mobile devices like smartphones have access to mobile broadband networks that allow you to make and receive phone calls.

- **Applications:** Applications meant for mobile devices are specifically designed for running on a particular OS. These applications are what extend the capabilities of devices beyond just connecting to the internet or making calls.

Other features generally found in mobile computing devices include GPS capability, accelerometer, compass, microphone, camera, and so on.

## Some common Mobile Computing Devices:

Mobile computing devices have evolved greatly over time. A lot of the devices that existed in the past have been phased out, like the Personal Digital Assistant (PDA). We live now in the age of the smartphone, but there are several other categories of mobile computing devices -

- **Laptop:** Laptop computers are portable personal computers. It is meant to offer the same functionality as a PC, so the same OS, applications, and files can run on this.
- **Smartphone:** A smartphone is a mobile phone with powerful capabilities. They typically have a touchscreen interface, have internet access, can run various applications, and include features like a camera and GPS.
- **Tablet computer:** Tablets have touch screens and virtual keyboards, and are often thought of as an intermediary between a laptop and a smartphone. They have better processing power, functionality, and screen resolution than smartphones. Some models may also have a stylus meant for navigating the touch screen better.
- **Wearable:** A more recent addition, wearable computers like smartwatches offer limited features similar to a smartphone within a watch.
- **E-reader:** E-readers are devices that are similar to tablets, but their main purpose is to read digital documents.

Other devices include scanners, carputers, handheld gaming consoles, advanced digital cameras, smart speakers, and so on.

## Characteristics of Mobile Apps

Mobile technology and mobile apps are the front runners of the digital revolution that change our way of living. As mobile technology advances and reaches new heights, so do mobile apps which fuel the so-called engine of mobile technology. As we move forward in this new era of mobile computing, mobile technology and apps are making its way into all aspects of our lives. Thousands of mobile apps are being developed every year to cater to our needs. There is one simple fact about mobile app and business. A great mobile app can definitely help your business to rise to the top. However, these days, a great mobile is not enough. With the thousands android apps and ios apps available in the app stores, your mobile apps need to not just be

great but be successful as well. A great mobile app does not translate to a successful mobile app, but all successful apps are definitely great apps. Regardless of your purpose in developing a mobile app, what industry you are in, or whoever your target audience is, it is important that your mobile apps have these five characteristics to make it successful.

**1. High and consistent performance:** The interface, high-tech features and functions of your mobile app will account to nothing if it cannot function well. High and consistent performance should be the first priority in developing the app. It should be well-tested and consistent in its performance even under extreme conditions. The ideal mobile app in terms of performance is the one that consumes minimum CPU and battery power, and does not require huge storage space. Loading time must be under 10 seconds. Generally, users get impatient and bored if startup takes too long. If a longer startup cannot be avoided due to huge amount of data that your app has to load, have a cool startup/loading screen that would let the user know that the startup process is taking place. High and consistent performance also mean minimal to zero crashes and freeze times. An app, however awesome, is doomed to be a failure if it keeps crashing and freezing. Reliability and consistency should be the first priority in developing a mobile app.

**2. Unique, Appealing and Easy to Use:** Your mobile app should be memorable and stand out from the rest. There are hundreds of thousands of mobile apps out there, the question is what will get your app noticed? Your mobile app needs to be amazing, pretty and appealing to your target audience. It must be uniquely packaged and fully branded. Also make sure that your idea is unique enough to be memorable, not just another imitation of some else's idea. Another key characteristic of a successful app is the ease of use. Generally, mobile users do not have the patience or time to learn how to use a complicated app, especially if your competitor is offering the same feature but with an easier to navigate interface. In designing an app, it's important to make sure that the user can get what they want quickly, if not instantly.

**3. Platform appropriate:** When designing and developing a mobile app, make sure that it is appropriate for a handheld mobile device. Remember that mobile technology has many different platforms, so you'll want to ensure that you have an appropriate design for all brands and devices. Don't limit the reach of your app by just developing an app for ios devices if you think it could also appeal to Android mobile users.

**4. Responsive customer support and regular update:** Expect that not all users of your app will be tech savvy. There will always be somebody who will need help on downloading and using your app. It is important that you respond to queries and requests promptly. Sometimes, one question can lead to a sale and are commendation or share in a social network. It is also important that your content is up-to-date and contains relevant information. Regular fixes to bugs and uploading new features is always a great way of making your end-users stay interested.

**5. Affordable:** Your app may be appealing and beautiful, but if it is not affordable, it is doomed to fail. Why should the customer choose your expensive app if he can get an app with same features for free? Try to offer both a free version and full-feature paid version. However, it is still important to have your full-feature paid version affordable.

## **Types of mobile application:**

To begin with, there are different types of Mobile applications:

### **1. Native Applications:**

These are applications developed to be used on a particular platform or operating system such as Android, iOS etc. Native apps are usually written in languages that the platform accepts. They are also built using the specific Integrated Development Environment (IDE) for the given operating systems, such as Android Studio for Android Apps and XCode for iOS Apps. The principal advantage of native apps is that they optimize the user experience. By being designed and developed specifically for that platform, they look and perform better. Examples of some popular Native Applications are Instagram for Android, VLC media player for Android, WordPress for iOS, Native Apps are usually built using either of the following languages;

1. Swift or Objective C for iOS applications
2. Java , Kotlin for Android applications
3. C# or VB.NET for Windows applications

**Pros** – They are very fast and responsive because they are built for that specific platform.  
– They have the best performance.  
– They are more interactive, intuitive and run much smoother in terms of user input and output.

**Cons**– They are considerably more expensive to develop compared to cross-platform and web applications.  
– They require more time to develop as one application has to be written in different languages for different platforms.

### **2. Web Apps :**

Web apps behave similarly to native apps but are accessed via a web browser on your mobile device. They're not standalone apps in the sense of having to download and install code into your device. They're actually responsive websites that adapt its user interface to the device the user is on. In fact, when you come across the option to "install" a web app, it often simply bookmarks the website URL on your device.

One kind of web app is the progressive web app (PWA), which is basically a native app running inside a browser.

**Technology Used:** Web apps are designed using HTML5, CSS, JavaScript, Ruby, and similar programming languages used for web work.

**Pros:** Because it's web-based, there is no need to customize to a platform or OS. This cuts down on development costs.

Plus, there's nothing to download. They won't take up space on your device memory like a native app, making maintenance easier – just push the update live over the web. Users don't need to download the update at the app store.

**Cons:** But this is also pertinent: web apps are entirely dependent on the browser used on the device. There will be functionalities available within one browser and not available on another, possibly giving users varying experiences.

And because they're shells for websites, they won't completely work offline. Even if they have an offline mode, the device will still need an internet connection to back up the data on your device, offer up any new data, or refresh what's on screen.

### 3. Hybrid Apps:

These are web apps that look and feel like native apps. They might have a home screen app icon, responsive design, fast performance, even be able to function offline, but they're really web apps made to look native.

**Technology Used:** Hybrid apps use a mixture of web technologies and native APIs. They're developed using: Ionic, Objective C, Swift, HTML5, and others.

**Pros:** Building a hybrid app is much quicker and more economical than a native app. As such, a hybrid app can be the minimum viable product – a way to prove the viability of building a native app. They also load rapidly, are ideal for usage in countries with slower internet connections, and give users a consistent user experience. Finally, because they use a single code base, there is much less code to maintain.

**Cons:** Hybrid apps might lack in power and speed, which are hallmarks of native apps.

# **Mobile app development frameworks**

Let's understand some of the top mobile app development frameworks, which make app development simple:

## **1. Swiftic**

It is one of the best mobile app development frameworks available in the iOS platform. It is featured with an easily navigable interface. Using Swiftic is a convenient option as you get 30 days money-back guarantee option for all its plans. Besides, it has another 6-months of success guarantee scheme. If you don't get any result from it, 6-month service is free.

Some of the significant features are:

- Interesting push notification
- Become a loyal shopper with a loyalty card
- Build your mobile store
- In-app coupons
- Use scratch cards to win prizes
- Easy Communication
- Menu & Ordering
- Customer Engagement
- App Promotion
- Social & Media Feeds
- App Publication Assistance
- Advanced Analytics
- Third-party integration

## **2. Native Scripts**



it is an open-source framework to create native mobile applications empowered with Angular, Typescript, JavaScript, CSS, and Vue.js.

Native Script is a preferable framework to reduce the code and time of the app loads on the system.

Many leading companies love using Native Scripts for its strong web empowerment platform.

Some of the Native Scripts features are:

- Native user interface without WebViews
- Full direct access to Android & iOS APIs
- Cross-platform application
- Mission-critical business support
- Robust backend support

### 3. React Native



It is the best JavaScript library to build native applications for all devices and platforms. With React Native, you can develop rich applications for both iOS and Android. It also allows creating platform-specific versions of various components allowing easy using of single codebase across various multiple platforms. This community-driven JS library was introduced by Facebook in 2018.

Some of the React Native features are:

- Low-code
- Compatible third-party plugins
- Declarative API for predictive UI
- Supports iOS and Android



### 4. Xamarin

It is one of the top mobile app development frameworks and it is .Net based. This native platform is introduced by Microsoft. Being a cross-platform and open source app building platform, Xamarin offers a development ecosystem with backend, API, components, etc. It is a .Net developer platform supported by various tools, libraries, and programming languages. With Xamarin developers, you can build native applications with the support of an active community for Android, tvOS, watchOS, iOS, macOS, and Windows.

Some of the Xamarin features are:

- A strong community of 60,000 contributors
- Versatile backend infrastructure
- Diagnostic tools
- Application loader
- Android SDK manager
- Storyboard files
- Google emulator manager





## 5. Ionic

Ionic helps you to build interactive hybrid and Progressive Web Applications along with cross-platform applications. This open source framework offers premium services for creating applications. Ionic covers building applications for the web, Android, and iOS. Besides, while working in Ionic, you can constantly create applications and ship them to the deployable locations. Ionic Studio is the lightning version of Ionic and it is quite powerful. You can install it locally to have an easy visual development environment.

With its out of box features, ionic is always seen as ideal platform over the other mobile app development frameworks.

Some of the Ionic features are:

- Intuitive UI components
- Faster development time
- Powerful & stable development platform
- Evergreen community of 5 Million developers
- Complete control over app building



## 6. Sencha Ext JS

Earlier known as Sencha, it was merged with Ext JS and now the new face is Sencha Ext JS. You can use this JavaScript framework for building data-intensive applications for the web as well as mobiles. It is also a cross-platform mobile development framework. Ext JS itself has more than 115 pre-integrated and high-performing UI components. Some of the components are HTML5 calendar, pivot grid, grids, trees, D3 adapter, lists, menus, forms, toolbars, windows, panels, etc. And Sencha has around hundreds of user extensions.

Some of the Sencha Ext JS features are:

- Can manage millions of data
- Powerful analytics
- Flexible layout system
- Visual data representation
- State-of-the-art Data-Driven Documents package (D3)

## 7. Onsen UI



Onsen UI is recognized as the most powerful tool when it comes to crafting complex mobile applications. It builds brilliant HTML apps with JavaScript, HTML, CSS. It also supports the libraries like Angular JS1, Angular 2+, React, and Vue. One thing that is applaud worthy of this platform is its competency to the app's appearance depending on the platforms. For instance, if we wish to have a native look for iOS and Android, it utilizes the automatic styling on the UI components. Its rich set of UI components includes tabs, stack navigation, lists, forms, etc.

Some of its significant features are,

- Zero time set up
- Simple to use and easy to learn
- Time-effective and cost-effective development



## 8. Flutter

Flutter is a UI toolkit by Google help in building native applications for the web, mobile, and desktop. This UI toolkit is featured with fully customized widgets, which support to create native applications in a short period. Besides, the Flutter's layered architecture ensures a faster rendering of components. Some of the striking Flutter features are:

- Built-in material design
- Built-in Cupertino (iOS-flavor) widgets
- Rich motion APIs
- Supports both iOS & Android
- Strong widget support
- High-performance application



## 9. Corona

Note: Corona is now also referred as Solar2D

Corona is a cross-platform & open-source development framework, which is suitable for building games and applications. It is also known as the 2D game engine. With Corona, you can make application for both mobile and desktop. This platform supports iPhone, iPad, tablets, Amazon Fire, Android phones, Mac Desktop, Windows Desktop, etc. Besides, you can build compatible applications for Fire TV, Apple TV, and Android TV.

Some of the striking Corona features are:

- Lua-based platform, a powerful & lightweight scripting language for games
- Multiple plugins for all needs
- Strong API support to integrate with any native library
- Faster development process
- Exceeding 500,000 Corona developers



## 10. JQuery Mobile

jQuery Mobile is a user interface system based on HTML5. This web is touch-optimized and used to build highly responsive applications for smartphones, tablets as well as desktops. jQuery mobile has ThemeRoller offering highly customizable themes. Besides, this top mobile app development framework enables you to create highly branded websites.

Some JQuery features are:

- Built on JQuery Core
- Lightweight size
- Configured with HTML5
- Automatic initialization
- Powerful theming framework
- Simple API



## 11. Mobile Angular UI

The Mobile Angular UI is the magical fusion of HTML5, Bootstrap and Angular JS. If you already know Twitter Bootstrap and Angular JS, you can learn Mobile Angular UI in no time. This mobile UI framework offers the basic mobile development components that are missing from Bootstrap 3. Some of these components are overlays, switches, sidebars, scrollable areas, etc. The fastclick.js and overthrow.js are the prime libraries of Mobile Angular UI.

Some of the Mobile Angular UI features are:

- Build HTML5 hybrid desktop & mobile apps
- No jQuery dependencies
- Build fully responsive interfaces with a super-small CSS file

### *Key Take Away*

After considering various factors like operating system, development time and much more, we have curated this list of top mobile app development frameworks.

From a single platform application to cross-platform application, mobile application development is evolving every day. With such evolution, the mobile development trends have witnessed a major paradigm shift in terms of code, script, platforms, integration, development method, and deployment. While we are getting into a hybrid, native, and progressing web applications, technologies like IoT, AI, ML, Blockchain are adapted to give the smartest, secure, and better user experience. It will be interesting to see how hybrid our applications will become with such fast-growing technology adaption.

## Android architecture

Android architecture contains different number of components to support any android device needs. Android software contains an open-source Linux Kernel having collection of number of C/C++ libraries which are exposed through an application framework services.

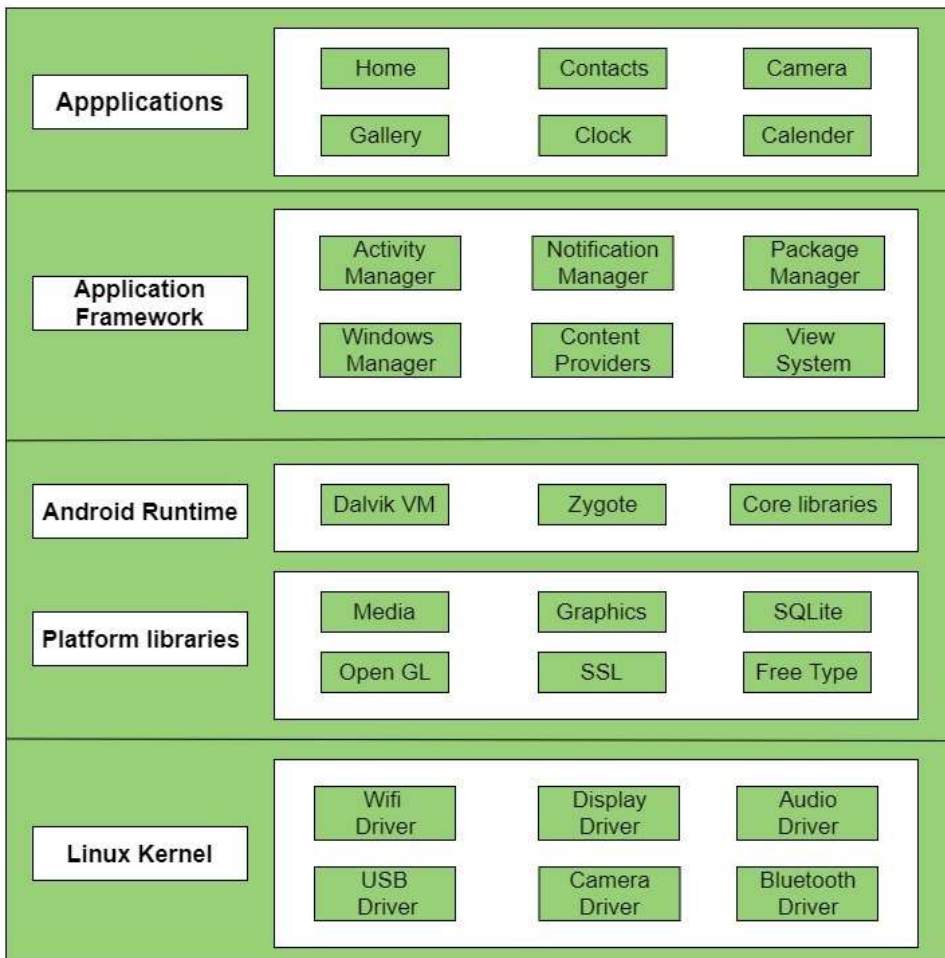
Among all the components Linux Kernel provides main functionality of operating system functions to smartphones and Dalvik Virtual Machine (DVM) provide platform for running an android application.

The main components of android architecture are following:-

- Applications
- Application Framework
- Android Runtime
- Platform Libraries

- Linux Kernel

Pictorial representation of android architecture with several main components and their sub components –



### Applications –

Applications is the top layer of android architecture. The pre-installed applications like home, contacts, camera, gallery etc and third party applications downloaded from the play store like chat applications, games etc. will be installed on this layer only.

It runs within the Android run time with the help of the classes and services provided by the application framework.

### Application framework –

Application Framework provides several important classes which are used to create an Android application. It provides a generic abstraction for hardware access and also helps in managing the user interface with application resources. Generally, it provides the services with the help of which we can create a particular class and make that class helpful for the Applications creation.

It includes different types of services activity manager, notification manager, view system, package manager etc. which are helpful for the development of our application according to the prerequisite.

### Application runtime –

Android Runtime environment is one of the most important part of Android. It contains components like core libraries and the Dalvik virtual machine(DVM). Mainly, it provides the base for the application framework and powers our application with the help of the core libraries.

Like Java Virtual Machine (JVM), **Dalvik Virtual Machine (DVM)** is a register-based virtual machine and specially designed and optimized for android to ensure that a device can run multiple instances efficiently. It depends on the layer Linux kernel for threading and low-level memory management. The core libraries enable us to implement android applications using the standard JAVA or Kotlin programming languages.

### Platform libraries –

The Platform Libraries includes various C/C++ core libraries and Java based libraries such as Media, Graphics, Surface Manager, OpenGL etc. to provide a support for android development.

- **Media** library provides support to play and record an audio and video formats.
- **Surface manager** responsible for managing access to the display subsystem.
- **SGL** and **OpenGL** both cross-language, cross-platform application program interface (API) are used for 2D and 3D computer graphics.
- **SQLite** provides database support and **FreeType** provides font support.
- **Web-Kit** This open source web browser engine provides all the functionality to display web content and to simplify page loading.
- **SSL (Secure Sockets Layer)** is security technology to establish an encrypted link between a web server and a web browser.

### Linux Kernel –

Linux Kernel is heart of the android architecture. It manages all the available drivers such as display drivers, camera drivers, Bluetooth drivers, audio drivers, memory drivers, etc. which are required during the runtime.

The Linux Kernel will provide an abstraction layer between the device hardware and the other components of android architecture. It is responsible for management of memory, power, devices etc.

The features of Linux kernel are:

- **Security:** The Linux kernel handles the security between the application and the system.
- **Memory Management:** It efficiently handles the memory management thereby providing the freedom to develop our apps.
- **Process Management:** It manages the process well, allocates resources to processes whenever they need them.
- **Network Stack:** It effectively handles the network communication.
- **Driver Model:** It ensures that the application works properly on the device and hardware manufacturers responsible for building their drivers into the Linux build.

# **Factors in Developing mobile applications**

## **1. Research**

If you have a great idea to develop a world class mobile application for your business, the first thing you need to understand is the market, customer demand and trends. Even before giving a technological touch to your business app, you need to do market research. A market analysis report can give you very useful insights. You can understand the popularity of similar apps available in the market and your competitor strategies. This will allow you to optimize your app from the very beginning. Apogaeis believes in “Learning from others mistakes than Learning after making mistakes.”

Customer reviews can provide you a glimpse of customer likes and dislikes, choice and preference and future demands. You can take a note of the pain areas and try to resolve those in your mobile app. Research can give you an opportunity to plan better and allow you to prepare a robust application from the very beginning.

## **2. Identify Target Audience**

This comes under research stage and extremely important. Identifying the right set of audience for your application is very important. The entire application future depends on these target set of users, as they play a very significant role in application development, as well as app feature expansion and growth. Questions like who are going to use my application and how it can add value to their lives, should be asked before starting development. If you meet the expectation of your users, definitely your app is going to get popular and it will help you to generate more revenue.

## **3. Right Platform Selection**

One of the most important question you need to ask yourself is, in which platform you want to deploy your business app. It is advisable to start with one platform. Be a master of a single platform, later you can move on other platforms. iOS, Android and Windows are the most popular mobile app platforms.

In order to decide the best-suited platform for your mobile application, you need to keep certain things in your mind like app brand, target audience, app features and most importantly pricing strategy. After that, you need to choose development methodology for your business app; Native, Mobile Web or Hybrid. Native applications have many advantages, but it may be slightly expensive.

## **4. Set Plan of Action**

Before starting to build your business mobile application, make sure you understand the entire process. A robust business app takes a decent amount of time and effort. It also passes through various stages. The business owner should understand the importance of all the stages, before going ahead. Project Management, App Design, App Architecture, App Development Methodology, App Testing, Enhancement and finally App Deployment- These are the basic stages of any mobile application. Set a plan of action for your app development process. Monitoring and controlling are very much essential. Release a beta version of your app first, and do rigorous testing and review before releasing the full version to the end users.

## **5. Know your Budget**

Developing a business mobile app requires investment along with knowledge and strategy. Knowing your budget and allocating it properly among each stage of app development is essential. There are various aspects of an app which requires money to be invested in; app development, maintenance, updating, marketing etc. Basically your app budget depends on your niche. The type of app you want to create and the type of content you wish to add in your app.

## **6. Think Out of the Box**

Always keep one thing in mind. People want something different. Why they choose one brand ahead of another similar brand? It is only because they feel the chosen brand has something unique to offer. Mobile applications are the best way to engage your customers and probably the easiest way to convert a lead into a potential buyer. Human beings have a tendency to get bored of one thing very quickly. As there are millions of smartphone applications available in the market, app users want to be served with something new. Hence, it is recommended to think out of the box strategies or features and don't let your users choose someone else.

## **7. Smooth and Efficient**

No excuse, your business mobile app must be smooth and efficient. If your app is taking a lot of time to load, it may prove to be fatal. Either user uninstalls the app after first use or rarely opens. It adds to create a negative impression in the user's mind. Make sure your app does not consume a lot of memory space and processing power on a mobile device.

Efficiency is something users expect from all the mobile applications. It includes user-friendliness, data efficiency, battery usages, security and alike. If your app is using a lot of 3G or 4G data, it might not work for you in a longer run. Users may download and forget. So, don't fall into that category and develop your app in such a manner that it won't consume more data of your users and does not drain battery.

## **8. User Experience**

User Experience is the backbone of any application. If you failed to provide your users excellent experience, then there is no reason for them to come back to you and use your app. User experience is an increasingly crucial feature when it comes to the digital landscape. An app defines how a user feels and think about your business and services. It is about making something valuable, easy to use and effective for your target audience. Don't make your app confusing or complicated.

A well thought mobile application offering seamless user experience can translate into several key benefits. The functionalities and content that are delivered to your app must be in line with your target audience. The end product must provide a superior quality customer service. If you are unable to provide a quality mobile application, it may harm your brand image.

## **9. Focus on Marketing Strategy**

There is no meaning of building an app for your business if you are not focusing on, how to market it and make it available to potential users. Creating a buzz, before the launch can promote your app in such a way that your app can get a lot of reach from the word go. Most of the business owners fail in executing their marketing strategies for their mobile application.



Different Mobile Applications have different targeted users. Similarly, you have to analyze your industry and your potential user base. After analyzing that only, you can make strategies on how to promote it. Online campaigns are one of the preferable options for many business owners. But your marketing strategy depends largely on your industry, offerings, and your potential users.

## **10. Testing**

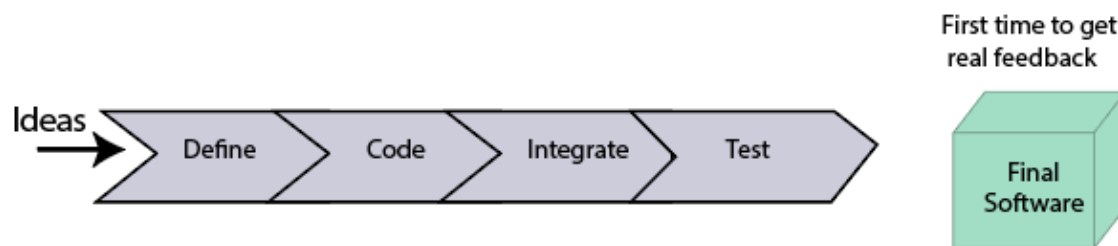
It is important to test your mobile application before you launch. Before users getting an experience, you should make sure it can fulfill their requirements. Smooth, efficient, high performance is few of the things you should keep in mind while testing. Testing will allow you to dust out all the nooks and crannies and make your app a strong business tool which can connect your business with your customers.

## Mobile Software Engineering:

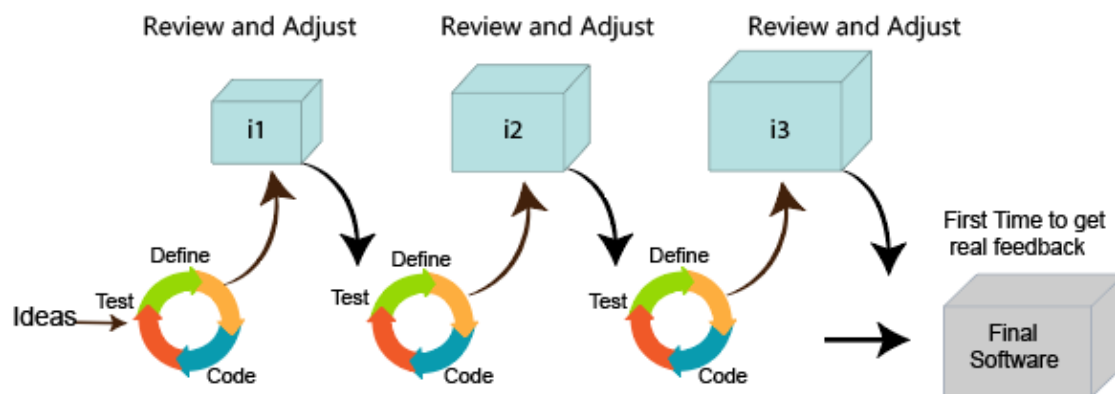
An agile methodology is an iterative approach to software development. Each iteration of agile methodology takes a short time interval of 1 to 4 weeks. The agile development process is aligned to deliver the changing business requirement. It distributes the software with faster and fewer changes.

The single-phase software development takes 6 to 18 months. In single-phase development, all the requirement gathering and risks management factors are predicted initially.

The agile software development process frequently takes the feedback of workable product. The workable product is delivered within 1 to 4 weeks of iteration.



Traditional Method



Agile Method

### **Roles in Agile**

There are two different roles in a Agile methodology. These are the Scrum Master and Product Owner.

## **1. Scrum Master**

The Scrum Master is a team leader and facility provider who helps the team member to follow agile practices, so that the team member meets their commitments and customers requirements. The scrum master plays the following responsibilities:

- They enable the close co-operation between all the roles and functions.
- They remove all the blocks which occur.
- They safeguard the team from any disturbances.
- They work with the organization to track the progress and processes of the company.
- They ensure that Agile Inspect & Adapt processes are leveraged correctly which includes
  - Planned meetings
  - Daily stand-ups
  - Demo
  - Review
  - Retrospective meetings, and
  - Facilitate team meetings and decision-making process.

## **2. Product Owner**

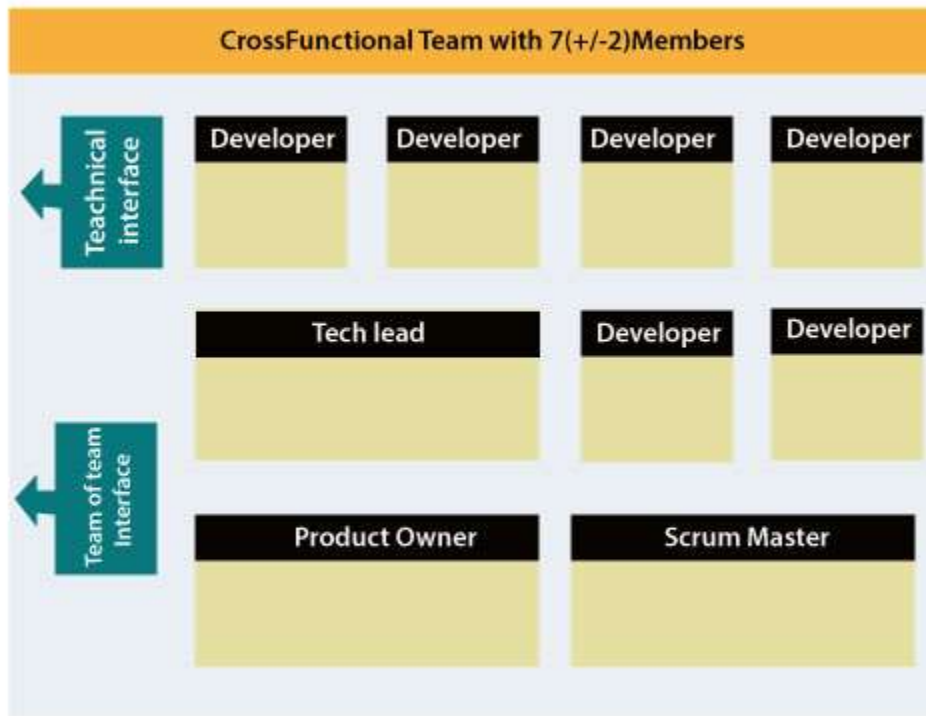
The Product Owner is one who runs the product from a business perspective. The Product Owner plays the following responsibilities:

- He defines the requirements and prioritizes their values.
- He sets the release date and contents.
- He takes an active role in iteration and releasing planning meetings.
- He ensures that the team is working on the most valued requirement.
- He represents the voice of the customer.
- He accepts the user stories that meet the definition of done and defined acceptance criteria.

## **Cross-functional team**

Every agile team contains self-sufficient team with 5 to 9 team members. The average experience of each member ranges from 6 to 10 years. The agile team contains 3 to 4 developers, 1 tester, 1 technical lead, 1 scrum master and 1 product owner.

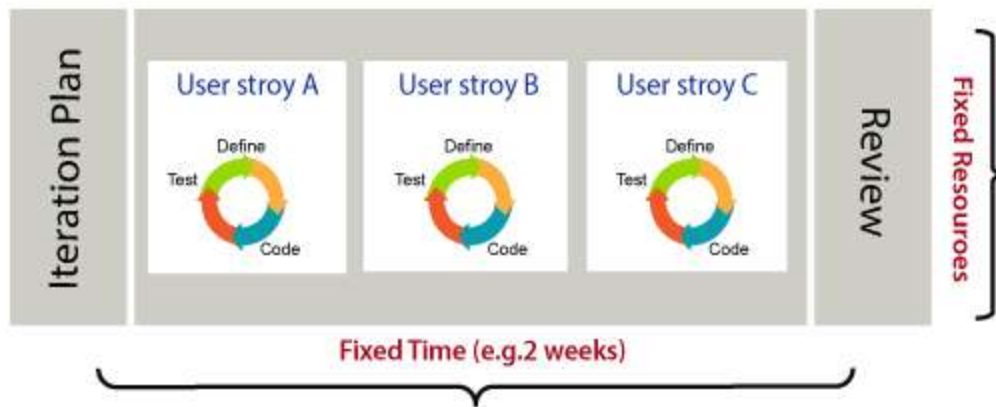
The Scrum master and Product owner are considered as a part of Team Interface, on the other hand remaining members are the part of Technical Interface.



### How an Agile Team plan their work?

An Agile methodology is not a specific set of ceremonies or specific development techniques. Rather, it is a group of methodologies that demonstrate a commitment to tight feedback cycles and continuous improvement. An Agile team works in iterations to deliver the customer requirement, and each iteration takes 10 to 15 days. However, the original Agile Manifesto didn't set the time period of two-week iterations or an ideal team size.

Each user requirement is a planned based and their backlog prioritization and size. The team decides, how much scope they have and how many hours available with each team to perform their planed task.



## What is a user requirement?

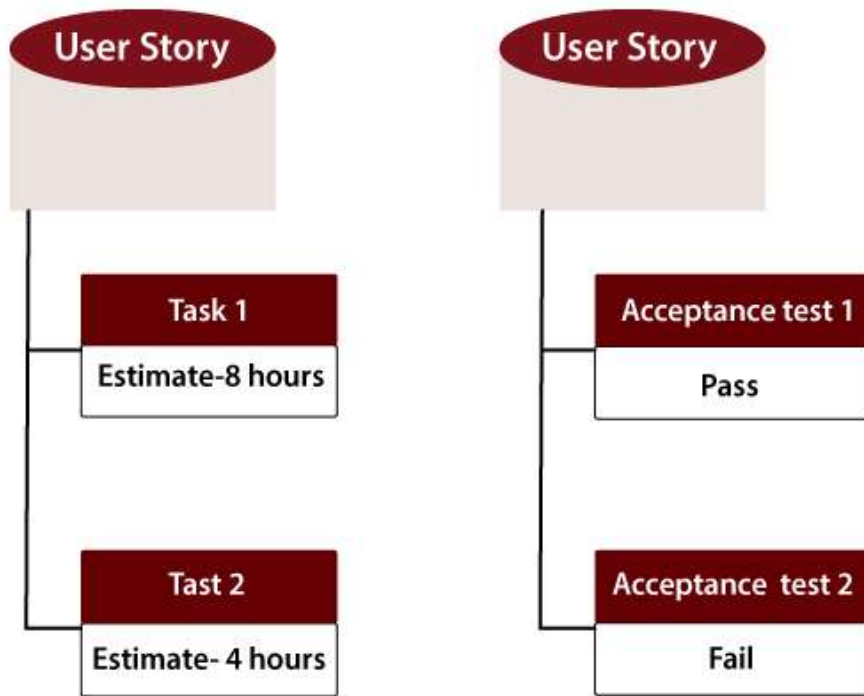
The user requirement defines the requirements of the user in terms of functionalities. There may be of two type of functionality.

- As a <User Role> I want <Functionality> so that <Business Value>
- In order to <Business value> as a <User Role> I want <Functionality>.

During software release planning, a rough estimate is given to a user requirement using relative scale points. During iteration planning, the requirement is broken down into tasks.

## Relation between User requirement and Task

- User requirement talks about what is to be done. It defines the needs of users.
- Task talks about how it is to be done. It defines how functionality is implemented.
- User requirements are implemented by tasks. Every requirement is gathering as the task.
- User requirement is divided into different tasks when it is planned in current iteration.
- User tasks are estimated in hours based, generally it is between 2 to 12 hours.
- Requirements are validated using acceptance test.



When the requirement is completed

The Agile team decides the meaning of task done. There may be different criteria for it:

- When the entire task (development, testing) are completed.
- When all the acceptance tests are running and are passed.
- When no defects found.
- Product owner has accepted the requirement.
- When the software product is delivered to the end user.

## User Interface (UI)

A good User Interface (UI) focuses on making user's interactions simple and efficient. User would appreciate a website with intuitive user interface that leads them towards their task in most engaging way. User Interface (UI) design focuses on thinking of a user, what they might need to do when they visit website and ensure that the interface has elements that are easy to access and understand. Being a UI designer, one need to understand the goals, skills, preferences and tendencies of the user to make a better interface.

The Generic User Interface (Generic UI, GUI) framework allows you to create UI screens using Java and XML. XML is optional but it provides a declarative approach to the screen layout and reduces the amount of code which is required for building the user interface.

The application screens consist of the following parts:

- **Descriptors** – XML files for declarative definition of the screen layout and data components.
- **Controllers** – Java classes for handling events generated by the screen and its UI controls and for programmatic manipulation with the screen components.

The code of application screens interacts with visual component interfaces (VCL Interfaces). These interfaces are implemented using the Vaadin framework components.

- **Visual Components Library** (VCL) contains a large set of ready-to-use components.
- **Data components** provide a unified interface for binding visual components to entities and for working with entities in screen controllers.
- **Infrastructure** includes the main application window and other common client mechanisms.

A screen is a main unit of the generic UI. It contains visual components, data containers and non-visual components. A screen can be displayed inside the main application window either in the tab or as a modal dialog.

The main part of the screen is a Java or Groovy class called **controller**. Layout of the screen is usually defined in an XML file called **descriptor**.

In order to show a screen, the framework creates a new instance of the `Window` visual component, connects the window with the screen controller and loads the screen layout components as child components of the window. After that, the screen's window is added to the main application window.

A fragment is another UI building block which can be used as part of screens and other fragments. It is very similar to screen internally, but has a specific lifecycle and the `Fragment` visual component instead of `Window` at the root of the components tree. Fragments also have controllers and XML descriptors.

A screen controller is a Java or Groovy class that contains the screen initialization and event handling logic. Normally, the controller is linked to an **XML descriptor** which defines the screen layout and data containers, but it can also create all visual and non-visual components programmatically.

All screen controllers implement the **Frame Owner** marker interface. The name of this interface means that it has a reference to a frame, which is a visual component representing the screen when it is shown in the main application window. There are two types of frames:

- **Window** – a standalone window that can be displayed inside the main application window in a tab or as a modal dialog.
- **Fragment** – a lightweight component that can be added to windows or other fragments.

Controllers are also divided into two distinct categories according to the frames they use:

- **Screen** – a base class of window controllers.
- **ScreenFragment** – a base class of fragment controllers.

Most of these user interactions are touch-based and happen on colorful touch screen displays that are bursting with high-level interactions. Naturally, basic mobile UI design principles differ from those of a traditional desktop UI. After all, users are, by definition, on the move; control is limited, giving new meaning to the phrase 'all thumbs. Actions and information need to be big, bold, clear, and simple.

As mobile adoption continues to rise year-by-year, it's time to develop a **mobile-first strategy**, embraced by the likes of Facebook or other social networks, who make sure their iOS and Android apps offer a polished user experience on hand-held devices. After all, when users have more choice and freedom to find mobile applications that work for them, a poor user experience can easily devalue your brand, hurt your revenue, and disengage your users.

Aside from investing in mobile applications, many ecommerce stores see the increase of purchases coming from mobile. If an online store doesn't optimize checkout experience, usability or their mobile app design, they may lose market share or even render themselves obsolete.

## **VUIs and Mobile Apps, Text-to-Speech Technique**

Voice user interfaces are user interfaces that are used through speech. Typical examples of voice user interfaces include smart speakers and voice assistants.

Voice UIs employ speech recognition and natural language understanding technologies to transform user speech into text and meaning. Speech is a tool for enhancing traditional touch user interfaces into multimodal voice user interfaces.

Voice user interfaces are highly intuitive as they use the most natural way for us to communicate: speech. They are significantly faster than typing to input information but significantly slower than reading or seeing to output information from the computer system back to the user.



The first voice user interfaces were IVR, Interactive Voice Response, systems that enabled users to interact with a phone system by using speech. Typically, IVRs recognized only digits, but nonetheless they were early voice user interfaces.

The first applications of voice UIs were interactive voice response (IVR) systems that came into existence already back in the 80s. These were systems that understood simple commands through a telephone call and were used to improve efficiency in call centers. Last time I encountered one was a customer support call center for home appliances where a synthesized voice asked me to say the name of the manufacturer of the device I needed help with. Once it recognized correctly my utterance “Siemens”, it directed me to a correct person.

Current voice user interfaces can be a lot smarter and can understand complex sentences and even combinations of them. For example, Google Assistant is perfectly fine with something like “Turn off the living room light and turn on the kitchen light”.

However, as these smart speakers always wait until the end of the user utterance and only then process the information and act accordingly, they will fail if the user hesitates with their speech or says something wrong.

## **Designing the Right UI**

In mobile app development, the UI (user interface) and UX (user experience) play the most significant roles in a mobile app’s success. From the perspective of a designer, an exceptional UI for mobile app can only be ensured when you follow some tried and tested UI design principles. It is always advisable to stick to certain principles that are accepted by the designers’ community. Such UIs not only attract the user, but also help in retaining them in the long run. Check the UI principles listed below:

- **Consistency of the design layout:** Different sections of the app must be coherent in the design, and there must be a consistent flow of the layout throughout the app. Apps like [Evernote](#), [Netflix](#), and [Dropbox](#) deliver great consistency in their design layout. This also applies to the common features of the apps built for the same OS.
- **Unambiguous Interactive Elements:** The design elements that are interactive in nature must be clearly depicted. Ambiguity should be avoided to make sure that the users are going to access such elements when required by them. Options hidden in menus may sometime be forgotten. Several surveys and studies have revealed that conspicuous menu options are not good for mobile apps. At least, the key navigation elements must be visible.
- **Single Trial Learning Experience:** The term ‘Intuitive’ is often used to describe mobile apps when suggesting that they are very user-friendly. Actually, this word means that users understand the flow of

the app without remembering the steps. Options should be placed in a manner where users can infer what the next step would be.

- **Anticipate and Answer Users' Queries:** When a user's tap a button, they want to know whether the process has started and how long will it take. As technology abstracts these actions, the users are usually kept in the dark while the process is going on. Use Toast in Android and notifications in iOS so users will be notified when a process has completed. The Gmail app is a good example of such a design, as it displays messages like "saved in draft, message sent, etc." to inform the users about the completion of the task.
- **Layered User Experience:** The UX should be layered. Layered here refers that all the features of an app must not be exposed at once. It should surface out as the users delve deeper into the app and discover new things. This would help in keeping the users' interest intact for a longer time. WhatsApp offers a good layered experience as users explore new features as they go deeper into the app.

Design an interface for a mobile app with more chances to succeed.

### 1. Uniqueness

There is no point in creating something that already exists, and it is the same with mobile app design. Your app should definitely have a number of advantages, but presenting them the right way is another story. Take Clear, for example, it is a to-do list, but its interface fundamentally differs from all other to-do lists. There are no buttons and there's no navigation menu, just tasks. Users can only navigate and interact with the app with the help of gestures, which was their unique feature. Uniqueness is one of the most important mobile design principles and that is why it stays on the first place in our list.

### 2. Structure

User interfaces for mobile applications should be organized in such a way that users will find it helpful and meaningful. Make sure similar items are grouped together and users won't spend tons of time trying to find what they need. You also need to keep in mind that the number of features and sections should not overwhelm the user, so be ready to sacrifice some less relevant features.

### 3. Contexts

Some apps can be used only on the go, others can be used in a relaxed atmosphere, and some apps can be used in both situations. Context seriously affects mobile user interface design, such as a bigger font, highlighted buttons, and only vital content. On the other hand, the interface can be entertaining and full of engaging options. Before moving on, you need to find out in what context your app will be used and then build the interface according to this information.

#### **4. Gestures**

Gestures allow users to navigate apps intuitively. Gestures can also add some gamification and may help you gain particular audiences, like teenagers. Mobile devices are designed for gestures, so try to get the most out of it and become a source of inspiration for other apps. Simultaneously, try use gestures that are familiar to your users and keep in mind how your users will be handling their device and in what context. Let's say if the user is holding the handrail on a public transport with one hand and trying to work an app with the other hand, it might be annoying if there are no other gestures but to pinch to expand, since this gesture requires two hands.

#### **5. Tolerance**

App interface design should be created in a way that if some options come up along the way, the system would suggest by default such variants that will be most beneficial to the user. Make sure your app allows users to undo certain actions and there are notifications showing that some changes will be made and can be undone. Tolerance means allowing users to make mistakes and giving them the opportunity to revoke changes.

#### **6. Consistency**

If there are several elements serving one and the same purpose in the app, they should look and function the same way. No exceptions. Changing one element means changing all other variants. If there are several styles of "submit" buttons on different pages, it can be frustrating and the app would look messy. Consistency means tidiness and it is certainly a must-have in every app, which is why in the majority of mobile app design companies there's a rule to create a UI kit for every project. This includes all elements of the app, and it becomes easier to follow the consistency principle with a UI kit.

#### **7. Communication**

The core idea here is to let users know your app registered the action and processing it even if it is not possible to give an immediate result. When the user performs a certain action, it can be considered an act of communication with your app, which is why the app should respond with something. Sometimes even the loader sign can be enough. From a psychological perspective, people need to get acknowledgment for their actions, and it is especially important in the banking sphere.

## **Multichannel and Multimodal UI**

One of the breakthroughs users can benefit of is that automatic speech recognition (ASR) improved highly significant over the last years. ASR now works good for dictation tasks. However, dictation is a highly specific use case which does not require the extraction of semantics from the utterances. Some applications use speech input for form filling. However, filling each single slot by speech is often not more efficient than typing. The question arises: What are important challenges in using speech as a “mainstream” modality? While ASR made significant efforts within the last years, e.g. partly driven by the successful application of deep neural networks, the identification of the intended semantic for a further processing by the dialog manager is still a rather difficult process. ASR capabilities are easy to integrate into new user interfaces by making use of available programming APIs. On the technical side one of the next challenges is therefore to realize conversational speech interaction in many applications. This requires to simplify the usage of NLP methods for information extraction, dialog processing and presentation, so that developers can easily deploy speech interfaces. Since the Internet is mobile nowadays and conversational speech is the most convenient interaction mode of complex applications that require more than simple gestures, this will enable even more services at the hand of the users. In that matter it is important to better understand the specific benefits that emerge for individual users. Information about these benefits can be revealed by observing the users’ modality choice behavior. Understanding the factors influencing users’ modality choice will enable interface designers to adapt applications to the advantage of the user, and to inform the user about extra possibilities of interaction.