JCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Volume-4, Special Issue-3, May 2016 E-ISSN: 2347-2693

Software Engineering Development process, user interface design, methods and tools for Mobile Application Development

Raghavendra Rao B G[1] and Vasantha S[2]

[1,2] Master of Computer Applications,
Sir M Visvesvaraya Institute of Technology, India

Available online at: www.ijcseonline.org

Abstract- This paper provides an overview of important software engineering development process, user interface design tools, methods, application portability, quality and security, cost of development, hardware, software, licenses and developer accounts, proximity, Embrace minimalism.

Keywords- Mobile devices, application development, software engineering, programming environments, user interface design, proximity, Embrace minimalism, closure,figure and ground and similarity.

1. INTRODUCTION

In recent years, mobile devices have gained popularity due to lower costs, small and sleek sizes, and the capability to act as a computer with you at all times. The increased use of mobile devices has created new issues for developers and network administrators, such as how to secure the devices, how to deal with increases in bandwidth, and how to make existing codebases usable on a device ten times smaller than it was designed for this book discusses these problems and many more, with a detailed overview of how to get started developing for a variety of mobile devices. If you are reading this, you are interested in learning about mobile development; we hope to give you the information and tools to start down the best path to develop a mobile application.

This paper is for any developer or team that needs to create, refine, or strengthen their mobile development strategy. From a development team of one to two people to an enterprise-level team with multiple divisions, the topic of mobile development will eventually come up. The problem is that mobile development is an animal all its own. There is a wide array of platforms, languages, features, and dimensions, and each has its own idiosyncrasies.

This paper will highlight those issues, and give examples for approaching and working with them. Specifically this paper shows you how to develop an application that connects to a remote service and implements device-specific functionality. The paper also explains the how and the whys and wherefores of mobile application development. But first, this paper assumes you're here for one of several reasons.

WHY YOU MIGHT BE HERE

As a developer in a competitive market, the following thoughts have almost surely crossed your mind, or they may have been brought to your attention by your managers:

i. Your competitors have mobile apps, but you don't.
ii. Mobile apps make good business sense.
iii. Your services would add value to a user's mobile experience but your website isn't mobile friendly.
iv. Do you need a mobile application or a mobile website?

Competition

Do your competitors offer products or services that you do not? Is that why they have an app? Is that a market you want to expand into? If you are already in that market, can you add any features to an app that will have more draw than your competitors? Differentiate yourself by leveraging the technology your customers have available without making it a gimmick. For instance, you could offer location-based incentives: when a customer enters your premises you can have your application display a coupon, discount, or any current promotions. This leverages the device GPS, which isn't something you can get with just a mobile website.

Quality vs. Time to Market

Sometimes, a bad mobile application or website can be worse than no mobile app or website. The iTunes App Store is littered with cookie-cutter applications that wrap RSS feed data. Often these cookie-cutter apps lose all branding of a given company, and such applications can negatively impact your reach. Things to consider when looking at developing an app is that in the Android Market, users are given a grace period during which they can request a refund for the full purchase amount. You need to know what you want to deliver, and understand that the way you deliver it makes your customers — and potential customers — know that you are serious.

Legacy System Integration

This gets into enterprise-level development, which is discussed in this paper. This paper is explains how to use a newer technology, OData, to expose data in a very mobile-consumable fashion. This paper explain the pitfalls and caveats to mobile application deployment (as opposed to development"), and the limitations to overcome when developing inside the company intranet bubble.

2. SOFTWARE ENGINEERING AND MOBILE APPLICATION DEVELOPMENT PROCESS

2.1 Cost of Development

There are many costs associated with mobile application development. Each developer will need hardware and software to develop the applications on. The team will need devices to test the software on. And if you want to deploy your application to any public market, then your company will need accounts on the various markets (these often renew annually).

2.2 Hardware

To develop good mobile apps, you'll need an Intel-based Mac because, simply put, you won't be able to physically build the iOS implementation of your application without one. The nice thing about the Intel versions of Mac is that you can run Windows on them either virtually (using something like Parallels, or VMWare Fusion) or on the bare metal (using Apple's BootCamp). Expect to spend between $800 (for a refurbished machine) and $1600 (for a brand-new machine).

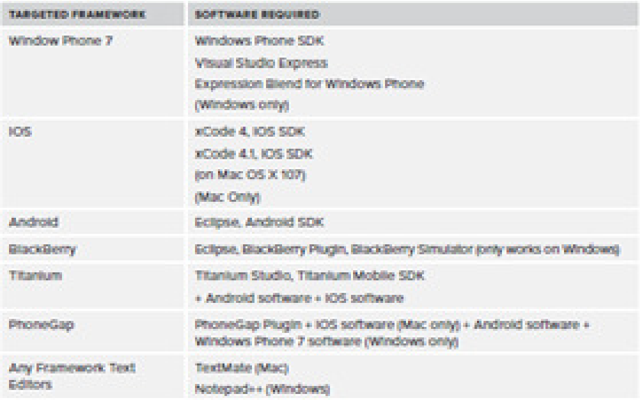
In addition to the Mac, you'll also need multiple monitors. When debugging any application, it is invaluable to step through your source while interacting with the running application. When developing, I have the emulator/simulator running in one monitor, My Dev Tool (IDE) running on another and a web browser on another with the documentation for the platform for which I am developing. Having access to all of this information at once prevents context switching for a developer, and helps maintain focus.

If you are seriously considering mobile development, you need to know that the emulator and simulators are great, but not perfect, so you'll need one of each of the types of devices you want to develop for. I can speak from personal experience: when developing an application, application behavior is not exact from the emulator to the device being emulated. This has happened to me on multiple platforms, so I cannot say that this is more prone to happen on one versus another. Here are some examples of devices you can use to test the various platforms as well as specific versions.

1) BlackBerry (6 or 7): BlackBerry Bold 9900
2) Android 2.2 (Froyo): Motorola Droid 2
3) Android 3.0 Tablet: Samsung Galaxy Tablet
4) Apple iPod Touch: iPod Touch 3rd Generation
5) Apple iPhone (versions 3.x and 4.x) (cell service): iPhone 3GS
6) Apple iPhone (versions 4 and greater) (cell service): iPhone 4
7) Apple iPad (WiFi or 3G for cell service testing): iPad 1
8) Apple iPad (with camera): iPad 2 or iPad 3
9) Windows Phone 7: Samsung Focus

2.3 Software

When developing mobile applications there are few overlaps when it comes to software. To develop for iOS you need a Mac, to develop for BlackBerry you need Windows, for Java-based frameworks use Eclipse. Building HTML for PhoneGap can be done in your text editor of choice. Table 1 and the following sections present an outline for what you will need for all of the platforms.

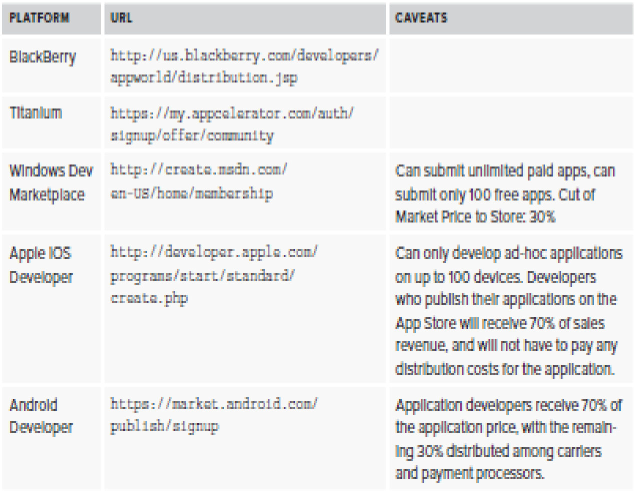


TARGETED FRAMEWORK	SOFTWARE REQUIRED
Window Phone 7	Windows Phone SDK Visual Studio Express Expression Blend for Windows Phone (Windows only)
IOS	xCode 4, IOS SDK xCode 4.1, IOS SDK (on Mac OS X 10.7) (Mac Only)
Android	Eclipse, Android SDK
BlackBerry	Eclipse, BlackBerry Plugin, BlackBerry Simulator (only works on Windows)
Titanium	Titanium Studio, Titanium Mobile SDK + Android software + IOS software
PhoneGap	PhoneGap Plugin + IOS software (Mac only) + Android software + Windows Phone 7 software (Windows only)
Any Framework Text Editors	TextMate (Mac) Notepad++ (Windows)

Table 1: Software Needed for Development

2.4 Licenses and Developer Accounts

The following table contains information regarding all of the various accounts necessary to develop for each platform and costs associated with such. In most cases you can expect to pay roughly $100 per platform annually for developer accounts.



PLATFORM	URL	CAVEATS
BlackBerry	http://us.blackberry.com/developers/appworld/distribution.jsp	
Titanium	https://my.appcelerator.com/auth/signup/offer/community	
Windows Dev Marketplace	http://create.msdn.com/en-US/home/membership	Can submit unlimited paid apps, can submit only 100 free apps. Cut of Market Price to Store: 30%
Apple IOS Developer	http://developer.apple.com/programs/start/standard/create.php	Can only develop ad-hoc applications on up to 100 devices. Developers who publish their applications on the App Store will receive 70% of sales revenue, and will not have to pay any distribution costs for the application.
Android Developer	https://market.android.com/publish/signup	Application developers receive 70% of the application price, with the remaining 30% distributed among carriers and payment processors.

Table 2: Licenses and Developer Accounts

3. A MOBIL USER INTERFACE DESIGN

Design falls into the category of craftsmanship: you do something until you are good at it, and then keep doing it until you are better. But many developers are too excited to solve the next functionality puzzle to spend much time with interface questions like appropriate color contrast or font. Don’t miss out on amazing design puzzles.

The latest generations of mobile devices are portable enough to carry at all times, connected to voice and data networks and contextually aware by using sensors and networks to preemptively complete tasks.

Current mobile limitations include bandwidth, times when users cannot access wireless Internet or phone networks, as well as a lack of technical capabilities, such as Flash, on many mainstream mobile devices. These constraints give application creators the opportunity to focus each application on a precise set of features. Mobile application creators can also use exciting new interactions with motion and gestures: zooming, swiping, tapping, turning, and shaking. These capabilities offer the chance to innovate.

Technology is changing and no device has a guaranteed market share in perpetuity, providing the easy excuse that the next device might change everything anyway. But like learning the syntax of one programming language and applying this knowledge to learn the next industry standard, good design transcends next season’s toy. Developers who understand the people who will use an application and the information users need will craft better applications no matter where technology goes next. So, let’s talk design.

This paper will introduce the mobile design context, tailing ways to use screen real estate efficiently. The rest of the discussion is divided among the people, the data, and the device. From Gestalt principles to accessibility on mobile devices, this paper covers understanding your users. A discussion of design patterns and content structure introduces mobile information design, using illustrations and real-world examples. An overview of platform-specific tips and resources ends the paper, with “Understanding Mobile Platforms.”

3.1 Embrace Minimalism

Limit the features available on each screen, and use small, targeted design features. Content on the screen can have a secondary use within an application, but the application designer should be able to explain why that feature is taking up screen space. Banners, graphics, and bars should all have a purpose.

3.2 Use a Visual Hierarchy

Help users fight cognitive distractions with a clear information hierarchy. Draw attention to the most important content with visual emphasis. Users will be drawn to larger items, more intense colors, or elements that are called out with bullets or arrows; people tend to scan more quickly through lighter color contrast, less-intense shades, smaller items, and text-heavy paragraphs.

A consistent hierarchy means consistent usability; mobile application creators can create a hierarchy with position, form, size, shape, color, and contrast.

3.3 Stay Focused

Start with a focused strategy, and keep making decisions to stay focused throughout development. A smaller file size is a good indicator of how fast an application will load, so the benefits of fighting feature creep extend beyond in-application user experience.

3.4 Understanding Mobile Application Users

While standing in line at the bank or a restaurant, people ull out their mobile devices to check in, entertain, and consume another dose of content. You can borrow metaphors from he real world, like a trash can or recycle bin holding deleted files; favor industry standards and make sure interface metaphors are appropriate to the device. Don’t be afraid to take new risks, but look to past design concepts to frame new ideas. The Gestalt principles have had a considerable influence on design, describing how the human mind perceives and organizes visual data. The Gestalt principles refer to theories of visual perception developed by German psychologists in the 1920s. According to these principles, every cognitive stimulus is perceived by users in its simplest form. Key principles include proximity, closure, continuity, figure and ground, and similarity.

3.5 Proximity

Users tend to group objects together. Elements placed near each other are perceived in groups; as shown in Figure 1, people will see one group of three gears, and one group of two gears. Many smaller parts can form a unified whole. Icons that accomplish similar tasks may be categorically organized with proximity. Place descriptive text next to graphics so that the user can understand the relationship between these graphical and textual objects.

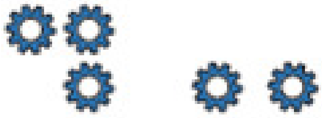


Figure 1: Proximity

3.6 Closure

If enough of a shape is available, the missing pieces are completed by the human mind. In perceiving the unenclosed spaces, users complete a pattern by filling in missing information. Figure 2 illustrates the concept of closure: people recognize a triangle even though the figure is not complete. Harness the closure concept to create icons with a strong primary silhouette, without overloading users on pixelated and overdone details. In grid patterns with horizontal and vertical visual lines, use closure to precisely show the inside and outside of list items.

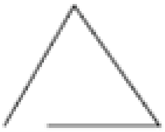


Figure 2: Closure

3.7 Continuity

The user's eye will follow a continuously-perceived object. When continuity occurs, users are compelled to follow one object to another because their focus will travel in the direction they are already looking. When people see Figure 4-3, they perceive the horizontal stroke as distinct from the curled stroke, even though these separate elements overlap. Smooth visual transitions can lead users through a mobile application, such as a link with an indicator pointing toward the next object and task.

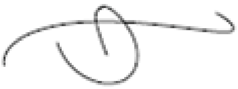


Figure 3: Continuity

3.8 Figure and Ground

A figure, such as a letter on a page, is surrounded by white space or the ground. In Figure 4-4, the *figure* is the gear icon, and the *ground* is the surrounding space. Complex designs can play with the line between "figure" and "ground," but mobile interfaces speed user frustration with unclear distinctions. Primary controls and main application content should maintain a distinct separation between figure and ground.

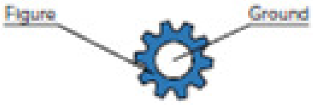


Figure 4: Figure and Ground

3.9 Similarity

Similar elements are grouped in a semi automated manner, according to the strong visual perception of color, form, size, and other attributes (see Figure 4-5). In perceiving similarity, dissimilar objects become emphasized. Strict visual grids confuse users by linking unrelated items within the viewport. The layout should encourage the proper grouping of objects and ideas.

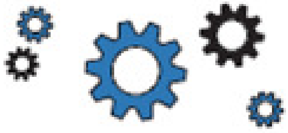


Figure 5: Similarity

3.10. Processes and Tools

As mobile applications become more complex and mission critical, development organizations must introduce processes that address more aspects of the development process than are covered in today's agile processes and development environments. As previously noted, the user experience is especially critical, so there is a greater need to create prototypes of the user interface(s), particularly when multiple devices will be supported. Testing is another important area for mobile software engineering research. One question involves the development of testing methods for product families, such as Android devices. It's insufficient to merely test an Android application on an emulator; it must be tested across many different Android devices running different versions of the operating system on various telecom networks, perhaps with l10n and i18n options. Integrated test suites would simplify this process.

Another area for research involves application maintenance in the rapidly changing world of mobile platforms. While "early adopter" consumers are often willing to update their device and their applications, most enterprise users are less likely to do so. In many cases, their companies will have policies discouraging them from doing so, as can be seen by the slow enterprise transition away from Windows XP and Office 2003. One particularly interesting question involves the use of virtualization technology on these devices as a way to support various platforms. Finally, application development and deployment is moving toward the "cloud". This new computing paradigm will not only affect development processes and tools, but also application architectures.

3.11 Portability

Application developers quickly developed apps for the iPhone platform following Apple's creation of the AppStore. As noted above, other providers of mobile platforms and devices have done the same (or are in the process of doing so). An important issue for the application developer is to decide which platform(s) to support in the highly fragmented world of mobile development.

Today, there are at least five important platforms (iPhone, Android, BlackBerry, Windows Phone, Symbian). From the standpoint of the application developer, it's quite expensive to support multiple platforms, especially when there are multiple versions and variants for each of them. The application developer has several options:

1) develop for a single platform only and use, to the extent possible, a common subset of the features available across all variants and versions of that platform; thus, for example, the developer would have only a single code base for an application that would run on different versions of the iPhone, the iPad, and possibly the iPod Touch. While that approach would simplify the developer's work, the resulting application would not be able to take advantage of all of the differentiating features of each device;
3) develop mobile web applications, thus minimizing the amount of native code for each platform; it remains uncertain whether this approach will meet the needs of the market, or;
4) use one or more layer(s) of abstraction that can map a "write once" application into native executable
programs that will run on multiple platforms. Each of these approaches presents a set of research questions, and suggests the need for customized tools to support cross-platform development and testing.

4. CONCLUSION

The items discussed in this paper are only a subset of the possible research topics in software engineering for mobile applications, but serve to indicate the breadth of research needs and opportunities in this emerging field. While the large number of mobile applications makes it appear that software development processes for them are well understood, there remain a large number of complex issues where further work is needed. In addition, there is a mobile "angle" to almost every aspect of software engineering research, where the characteristics of mobile applications and their operating environments present a new or different set of research issues

REFERENCES

[1] Agrawal, S. and A.I. Wasserman, "Mobile Application Development: A Developer Survey".

[2] Apple Developer Connection. http://developer.apple.com/iphone/index.action.

[3] Windows Phone developer site http://developer.windowsphone.com/windows-phone-7-series/ .

[4] Fring, Brian. 2009. *Mobile Design and Development*. O'Reilly.

[5] Schwaber, K. 2004. *Agile Project Management with Scrum*. Microsoft Press.

[6] Apple. iPhone Application Programming Guide.

[7] Reto Meier: Professional Android 4 Application Development, Wrox Publications 2012.

[8] David Mark, Jack Nutting, Jeff LaMouche, and Fredric Olsson, Beginning iOS6 Development:Exploring the iOS SDK, Apress, 2013.

[9] Charlie Collins, Michael Galpin and Matthias Kappler, Android in Practice, Dream Tech. 2012.