

# Addressing the Issues in Mobile Application Development

Shriya Gupta

Department of Computer Engineering, Shri Govindram Seksaria Institute of Technology and Science (SGSITS), India  
shriya\_mgupta@yahoo.com

[www.ijcseonline.org](http://www.ijcseonline.org)

Received: 17 June 2014

Revised: 10 July 2014

Accepted: 22 July 2014

Published: 30 July 2014

**Abstract**— As computer devices adopt new form factors and usability paradigms, mobility has become a byword in software development. The market is flooded with applications for the mobile devices and more and more developers are realizing the immense potential of this expanding market. This paper highlights some of the major issues faced by the developers of mobile applications. It also suggests a context-specific approach to address them.

**Keywords**— Application Development; Challenges and Issues in Mobile, Data Access; Maintenance in Mobile; Mobile Apps; Mobile Devices; Native and Cross-Platform Development Tools; Security in Applications; Software Engineering

## I. INTRODUCTION

As advances continue to be made in the field of mobile technology, developing mobile applications for handheld devices like mobile phones and Personal Digital Assistants holds immense potential. As per [1] illustrated in Fig. 1, Mobile use for ‘Messaging and Social Apps’ and ‘Utilities and Productivities Apps’ grew by 203% and 149% respectively in 2013, with the overall App usage growing by 115%. Research report [2] estimates that the market for application development services is likely to reach \$100 bn by 2015. This market evolution poses new challenges for developers to create applications which will fully exploit mobile’s new and expanding capabilities.

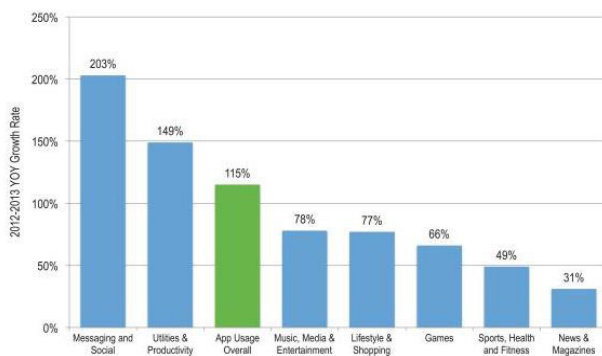


Fig 1. Mobile Usage Growth Rate in 2013

Developers are confronted with questions on selection of target platform, securing data access, enrichment of user experience, offline capabilities etc. They are also offered a lot of flexibility in the development environment – they have a vast array of tools for both native development and cross-platform development. Cross-platform development tools allow the usage of one programming language that can support multiple platforms as opposed to native development tools which are restricted to a single platform, often the

mobile platform’s own SDK. This paper addresses some of the key issues faced in application development.

## II. SECURITY

### A. Issues

Security is of paramount importance in any standard mobile application. More often than not, data transfer occurs wirelessly via GPRS across public networks. The data security in the process of passing data through network faces severe threat [3]. Additionally, loss of mobile devices is frequent. Security is therefore a primary responsibility of the application developer. Generally, each operating system has its own security implementation tailored to its own architecture. The degree of security support they provide also differs. This means that applications must arrange for their own security. Applications themselves differ in the level of security they must provide. A well-executed security strategy toes the line between a vulnerable application with insufficient security and a cumbersome application where users dislike multiple authentication checks.

### B. Approach

Encryption is a popular option for data security as it prevents data theft even when device is lost [4]. Other options that developers can invest in include binary hardening, and options to destroy data if user reports lost device and backup capabilities where data is archived on periodical basis. This can serve as a good strategy to overcome lack of transport security. Design must ensure that the security mechanism is consistent with the power capacity of device which may be very limited. Effort should be concentrated on most common and easily addressable vulnerability sources of applications, as concluded by research [5] from HP Security Research which tested mobile applications for security vulnerabilities –

**1. Privacy issues:** Of 2,107 mobile applications scanned, 97% accessed private data sources.

**2. Lack of binary protections:** 86% of applications tested lacked binary hardening. Applications were thus vulnerable

Corresponding Author: *Shriya Gupta*

to information disclosure, buffer overflows and poor performance.

**3. Insecure data storage:** 75% of applications did not use proper encryption techniques.

**4. Transport security:** 36% of applications tested sent user names and passwords over HTTP or incorrectly implemented SSL/HTTPS. Together with their web-application counterparts, these unprotected credentials leave application vulnerable to malicious attackers on the same network who can then sniff that data.

### III. USER-FRIENDLINESS

#### A. Issues

Mobile applications users often use their devices on-the-go which means that they require a great deal of simplicity in the User Interface (UI). They need to obtain specific information in time-effective manner. But with more people entering into the field of application development, multiple applications sometimes offer similar functionalities and this means that the application with the more appealing UI is preferred over the competition. Therefore, the developer needs to strike the right balance between simplicity and appeal. Trying to increase the appeal with better graphics and other trivial features also takes a toll on the device resources which slows down the application and makes it bulky. Mobile devices available today are very heterogeneous with regard to their display and input capabilities and used software platform configurations. The mobile UI paradigm is based around widgets, touch, physical motion, and keyboards (physical and virtual) rather than the familiar WIMP (Windows, Icons, Menus, Pointer) interface style of Apple's iOS and Microsoft Windows [6]. This leads to a complex development process for applications which have to target a wide range of devices [7]. Developers need to address the need for application to adjust to differing screen sizes and the option to toggle between portrait and landscape modes of display.

#### B. Approach

A crude way to test user-friendliness for different devices is by means of emulators which are virtual simulations of the actual devices. They are low-cost testing tools made available with the development kits for each platform but are often not accurately matched to the real devices [8]. A final test-run on real devices is still necessitated to ensure adherence to all non-functional requirements. Furthermore, the UI paradigm must not greatly deviate from the current trends. This prevents the user from the hassle of acclimatizing to a new application. Designs which seamlessly integrate themselves with the user's way of functioning considerably smoothen the reception of a new application.

### IV. DATA ACCESS

#### A. Issues

Where to store the user data is a question that application developers need to address before they begin the design phase. Possible options are user's own device file system and

online server. Both have their own pitfalls and advantages. While storage space in the user's device consists of an array of independent small form factor hard disks connected to a host by a storage interface in a mobile computing environment [9], it is advantageous for its all-time availability and possible time-efficiency, it is not a popular option on account of less and rigid storage space and its inability to cater to global access across devices. At the same time, back-end storage poses a problem when user is unable to connect to the network. The time taken for distributed data transfer is also a source of delay.

#### B. Approach

Ideally, an application should consider a hybrid solution wherein the application communicates with the online server at frequent intervals to synchronize and update the data which is cached on user device. This architecture requires built-in capabilities for transaction commit and roll-back to prevent data inconsistency from arising. Format of data storage should be chosen as per application requirements to optimize the performance of application. Objective is primarily to reduce overall look-up, transformation, translation and storage time of data.

### V. MAINTENANCE

#### A. Issues

Maintenance is largely a function of evolving with the user needs and changing trends. It is also a way of fixing bugs in applications. Apps differ significantly from traditional software systems in a number of ways, which calls for the development of novel approaches to maintain and comprehend them [10]. This necessitates a mechanism for developers to keep abreast of application usage and user satisfaction thereof.

#### B. Approach

This mechanism can be automated by means of analytic tools or solicited as user feedback. Analytics can monitor which application features are more frequented by users and identify their application usage patterns. Flurry Analytics, for instance, an industry standard in mobile, provides more than 125,000 developers the business data they need to understand their audience, usage and performance [11]. While developing analytical solutions, the developer needs to incorporate mechanisms that log offline activity and report it when user goes online. Analytics can also be a useful feature to resolve technical issues by keeping logs of crashes and other failures.

User feedback offers greater flexibility in user input as it does not require adherence to automatic testing tools but can be taken through several means like e-mail/telephone/form etc. Together these means provide developers with insight into the user needs and limitations of the application. Application maintenance, realized by updates, ensures that application evolves with the passage of time.

Fig 2. Mobile Application Maintenance: A Cyclic Approach



A cyclic approach to application maintenance is illustrated in Fig. 2 and it incorporates the guidelines presented in [12]. This approach is as follows –

**1. Retain Existing Users and Attract New Users:** This achieves the objective of the application development by leveraging the features that the application provides. This is a direct outcome of providing an improved feature set to users.

**2. Understand User Requirements:** Having an established user base dictates the action of understanding its requirements in order to fulfill them. This may involve automated analytic tools or user feedback.

**3. Plan for Upgrades:** After the identification of user requirements, it is important to plan for upgrades during the application development phase itself because feature improvement requires iterative usage of resources like budget, time etc. It is therefore, wiser to not spend all the resources during the development of first version of application. An alternative approach deals with allocating only part of resources to first time application development and following it up with allocation to upgrades. This is loosely based on the Agile Development Methodologies.

**4. Upgrades development, testing and release:** There exists two different categories of users in the context of upgrades –

a) *Users downloading the application the first time:* This class of users are not majorly concerned with any past version of the application and their expectations only run to achievement of desired future functionality from the application.

b) *Users with pre-installed application:* This class of users dictates greater caution when engineering upgrades. They require the upgraded application to be consistent with the previous version. This can be achieved by backing-up of existing application data in previous version and capabilities for forward engineering and reverse compatibility in order to support users who are unable to upgrade to the latest version at given time but may later upgrade to a newer version.

Thus, a migration tool is needed if application also caters to the latter class of users. Version control and release info must be judiciously implemented and documented to provide upgrade specifications. Testing needs to be undertaken irrespective of the class of users.

**5. Improve features over time:** Improving the features over time ensures that the application itself gets better with time. This is required to retain user interest and to garner new interest. Improving features regularly is also a testament of the developer's commitment to quality application development and this invokes user loyalty.

## VI. DEVELOPMENT TOOLS

### A. Issues

Developing for individuals platforms like Apple iOS, Android or Microsoft Windows Phone 7 is achieved by use of native development tools whereas cross-platform development tools allow for applications capable of running on multiple applications. Deciding which solution is better suited to catering to the user requirements can be perplexing.

TABLE I.

COMPARISON OF MOBILE DEVELOPMENT TOOLS

Mobile Device Platforms	Tools/Technology				
	Native OS	Native API	.NET	Java	Adobe Flash
Windows CE/Mobile	Windows CE	Yes eVC (C++)	Yes	Yes ME <sup>a</sup>	Yes
Blackberry	Blackberry	Yes	No	Yes ME	No
Android	Linux	Yes C,C++	No	Yes Dalvik VM	Announced HTC Hero
Symbian	Psion EPOC descendent	Yes Symbian C++	No	Yes ME	Yes
iPhone	iOS	Yes Obj. C	Yes	No	No
Palm webOS	Linux	No	No	No	Announced

a. ME – Micro Edition

Table 1 contains a comparison of the support for popular development tools [13] on different platforms.

### B. Approach

The choice of which tools to use is largely dictated by development budget, time constraint and scope of application objective. While developing for native platform can be costly for multiple platforms, the application is generally superior in functionality and performance as the design makes best use of the OS-specific tradeoffs at each level. The cross-platform tools strive for uniformity and for this reason are forced to make inferior tradeoffs for the sake of generalization. Facebook's iOS application used to be a HTML5 solution wrapped as a native application, but they released a completely new native iOS application [14] written in Objective-C in August 2012, citing "that when it comes to platforms like iOS, people expect a fast, reliable experience and our iOS app was falling short." Fig. 3 depicts the relationship between 'cost and complexity' and 'performance and functionality' in the context of different development

tools [15]. Here, the 'Web/hybrid Apps' make use of cross-development tools and 'Native with extensions' provides for greater control over low-level system resources like hardware. Despite its superiority, a major challenge faced by several developers in native development is that of stringent requirements for admission into public application stores and this keeps several developers from migrating to Native Development.

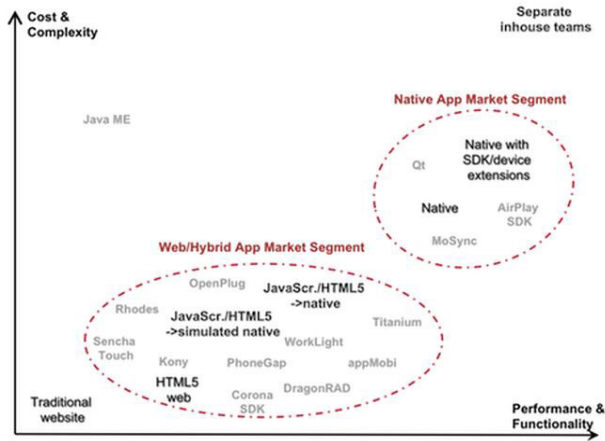


Fig 3. Illustrative Overview of Market Segment of Development Tools

## VII. CONCLUSION

It is a good strategy for developers to identify in the beginning itself the target platform for the mobile application. And with development toolkits being different, developers considering multiple implementations can benefit from adherence to common testing, project management and reporting tools. This facilitates both development and testing. It is essential to accommodate the mobile platform limitations like limited storage, less computational resources, low battery life etc. Application developers should leverage the mobility and cost-effective advantage of mobile.

Detailed analysis is needed to identify the right level of security and the means of implementing it. Otherwise, applications run the risk of vulnerability or cumbersome. Additionally, the User Interface of applications is a significant component of the user experience and must be simplistic in design with an appealing interface. This requires more effort than traditional UI solutions on account of greater restrictions imposed by mobile users. Data Access and Maintenance are best supported by hybrid models and Mobile Application Maintenance approach respectively.

Mobile application development requires being in touch with evolving best practices, such as [16, 17], in the industry. The issues faced in mobile application can be resolved in several ways. But the quality of solution employed makes the difference between a successful application and an unsuccessful one. Developers who have a deep understanding of the user application requirements, scope and audience will find it easier to judge the right trade-offs to be made at all stages of the development.

## ACKNOWLEDGMENT

The author wishes to thank Shri Govindram Seksaria Institute of Technology and Science for the opportunity provided to do research. The author also thanks Prof. Urjita Thakar for her support and guidance with doing research and all the professors of the Department of Computer Science, Shri Govindram Seksaria Institute of Technology and Science for their encouragement.

## REFERENCES

- [1] Simon Khalaf, "Mobile Use Grows 115% in 2013, Propelled by Messaging Apps", Flurry Analytics, January 13, 2014, Retrieved March 09, 2014 from <http://www.flurry.com/bid/103601/Mobile-Use-Grows-115-in-2013-Propelled-by-Messaging-Apps#.U21TC4GSxlc>
- [2] Research2guidance, The Mobile Research Specialists, "The Market For Mobile Application Development Services", July 2011
- [3] Xiandi Zhang, Feng Yang, Zhongqiang Liu, Zhenzhi Wang, Kaiyi Wang, "Research and Application of Data Security for Mobile Devices", International Federation for Information Processing, Advances in Information and Communication Technology, Volume 346, 2011, pp. 46-56
- [4] ABIresearch, Technology Market Intelligence, "Mobile Authentication & Encryption", Code AN-1325, Quarter 4, 2013
- [5] HP Security Research, Press Release, November 1, 2013, Retrieved Feb 09, 2014 from <http://www8.hp.com/us/en/hp-news/press-release.html?id=1528865#.U21Yc4GSxlc>
- [6] W. Anthony, "Software Engineering Issues for Mobile Application Development", FoSER 2010, Retrieved April 10, 2014 from [http://works.bepress.com/tony\\_wasserman/4](http://works.bepress.com/tony_wasserman/4)
- [7] T. Butter, M. Aleksy, P. Bostan, M. Schader, "Context-aware User Interface Framework for Mobile Applications", Proceedings of 27th International Conference on Distributed Computing Systems Workshops, ICCDCSW, 2007, pp. 39
- [8] Biraj Das, "How to test Mobile Applications – Emulators or Real Devices", Congruent Blog, Retrieved March 03, 2014 from <http://blog.congruentsoft.com/how-to-test-mobile-applications-emulators-or-real-devices/>
- [9] M. Nijim, Lee Young, L. Bellam, "Hybum: Hybrid Energy Efficient Architecture for Mobile Storage Systems", Proceedings of 9th International Conference on Information Technology: New Generations, 2012, pp. 214-220
- [10] R. Minelli, M. Lanza, "Software Analytics for Mobile Applications – Insights & Lessons Learned", Proceedings of 17h European Conference on Software Maintenance and Reengineering, 2013, pp. 144-153
- [11] Flurry Analytics, About Us, Retrieved March 15, 2013 from <http://www.flurry.com/about-flurry>

- [12] Serge Huber, "Tips for Mobile Application Maintenance", August 29, 2012, Retrieved April 13, 2014 from <http://www.aiim.org/community/blogs/expert/tips-for-mobile-application-maintenance>
- [13] Moochman, "Mobile OS Shootout: The Cross-Platform Developer POV", March 23, 2009, Retrieved April 13, 2014 from [http://www.osnews.com/story/21179/Mobile\\_OS\\_Shootout\\_The\\_Cross-Platform\\_Developer\\_Point\\_of\\_View](http://www.osnews.com/story/21179/Mobile_OS_Shootout_The_Cross-Platform_Developer_Point_of_View)
- [14] J. Dann, "Under the hood: Rebuilding Facebook for iOS", Retrieved Feb 02, 2013 from <https://www.facebook.com/notes/facebook-engineering/under-the-hood-rebuilding-facebook-for-ios/10151036091753920>
- [15] Jonas Lind, "Platform X: How Cross-Platform Tools can End the OS Wars", June 26, 2011, Retrieved March 09, 2014 from <http://www.visionmobile.com/blog/2011/06/platform-x-how-cross-platform-tools-can-end-the-os-wars/>
- [16] World Wide Web Consortium, Mobile Web Application Best Practices W3C Working Draft, 13 July 2010, Retrieved Jan 6, 2014 from <http://www.w3.org/TR/mwabp/>
- [17] M. Nagendra and B.Kondaiah, "A Comparison and Performance Evaluation of On-Demand Routing Protocols for Mobile Ad-hoc Networks", International Journal of Computer Sciences and Engineering, Volume-02, Issue-05, Page No (15-19), May -2014.