

Role and Impact of Software Engineering in Mobile and Wireless Application

Faheem Ahmad
Research Scholar
Shri Venkateshwara University
Gajraula, Uttar Pradesh, India

Shafiqul Abidin, PhD
Associate Professor
Department of Computer Science
Aligarh Muslim University
Aligarh, Uttar Pradesh, India

ABSTRACT

In this world, smart devices are infiltrating with such a high speed that we can find them everywhere. Because of the widespread usage of the smart devices, software developers and engineers must be aware of the fascinating characteristics of mobile phones and familiarize their formative methods to meet the requirement of the software used in mobile devices. Due to the simple application transferability, distinctive sensors, and high processing, the designers and developers of autonomous and commercial systems have established their new operational ground within the frame of agile devices. However, we discover that the computer programmers who are responsible for the fast-paced growth of this modern stage are required to update themselves. The paper will discuss four main challenges faced by mobile software engineers for a variety of applications, as well as potential research opportunities. We would also be examining at the different issues given: A) Creating a User Interface for disabled people B) Development of platform Independent Application C) Development of context-aware application, and D) To deal with ambiguous requirement.

Keywords

User-Interface, Context-aware, platform Independent, Software Product Line Engineering, Multi-agent system

1. INTRODUCTION

The popularity and use of smart phones are rapidly growing. It is estimated that more than 2 billion users are worldwide currently [1]. The rapid development of portable devices has a significant influence on the stages applied for the development of software used in the marketing, games, commerce, and social networking etc. With the voice recognition system, different type of web browsers, high quality of cameras, remote capabilities, and high accuracy of GPS, modern devices can provide their users with a large number of options through mobile applications [2]. Nevertheless, the availability of large number of advance quality of sensors poses a difficult task to today's computer software engineers, requiring them to perform tasks that were previously not expected from traditional application engineers [3].

Because of their effortless device transferability, special sensors, and computing power, designers of autonomous applications obtained their novel operating environment within the frame of portable gadgets. The program designers, on the other hand, are missing from the fast-paced creation of this unused level. The program development approach used in traditional development is not suitable for the development of modern mobile application. Essentially, the User Interface that furnish a fully untapped demonstration of the intelligence of the human being within the context of augmented reality,

QR filtering, and multi-touch, among other things, that has not been explored by analysts and for which no current rules have been developed [4]. The numerous stages that show inside the flexible gadgets, rendering them diverse in Operating Systems, equipment (Samsung and Apple etc.), or indeed the various formats (phones, tablets etc.), needs the creation of that software which can be used on a variety of devices. Lastly, the portable stages' individuality presents both a contest and a prospect. For example, Murphy, Roman and Picco, contend that "mobility represent a total meltdown of all the stable assumptions" which were previously fragment of traditional computer program development and development of software application [5].

Through this paper, we have tried to show how the above-mentioned components will pose four most important challenges encountered by developers during the development and engineering of the applications. These challenges come to the developers who are in charge of planning and developing a very advance and sophisticated applications for mobiles using the fantastic features, facilities furnished by mobile devices.

2. SOFTWARE DEVELOPMENT IN MOBILE

Now, we have focused the four main challenges encountered by computer programmer and software engineers for mobile devices, based on the three components discussed above.

Software engineers face huge challenges in designing and development of successful software to cater the need of uses. These are discussed below:

Table 1: Challenges in software engineering

Platform Independent	How to meet the requirements of different types of platforms
Complexity	How to effective produce different types of applications
Interoperability	How to work together with different sets of applications
Business Demands	How to develop a software quickly and cost effective to survive in the market
Future Requirement	How to deal with future requirements which are not available
Diversity	How to develop different type of applications for user requirements

2.1 Universal User Interface Development

We discuss a few key research projects that were undertaken

in the development of a universal user interface for mobile devices. Each stage show in the flexible world has its own unique approach to taking care of the developer's UI requirements. Under the UI laws, there are a few overlapping subjects.

The size of the screen and the size of the UI are the most essential considerations when creating a mobile user interface. Varied devices have different screen sizes, and Apple's resolution is limited to the iPad and iPhone formats. As a result, it is more difficult to prepare the user interface because the target devices must be considered well in advance. Since their presentation, Shneiderman's 8 genius rules for interface design have gained recognition and ubiquity [6]. However, they are not well-suited for use in a variety of devices. As per the research done by Gong Tarasewich, four of Shneiderman's eight rules can be directly linked to mobile devices, such as: backing inner control loci, planning dialogs that compromise closing, providing instructive feedback, and allowing visit clients to use alternate routes. The remaining laws must be adapted to accommodate the diverse world [7].

2.2 Platform Independent software

The platform independent encompasses a variety of computing situations e.g., smart phones and tablets, providing techniques, different vendors like Samsung, Apple, Nokia, Microsoft, and different Operating Systems iOS, Android and window etc. of these variables must be carefully considered when developing any mobile application, since they will directly affect the requirement of application's requirements. These vendors are hoping to plan for focusing on one stage for portable gadgets which are full of attribute, or portable websites that can appeal to numerous stages, or even concentrate on a series of portable gadgets with rich inbuilt applications in today's setting. When designers are aiming for a single application, they must either create an only one application for all stages at the expense of a number of function differentiation or create multiple adaptations of the same application that are appropriate for various computing / equipment stages [8].

Due to the current circumstances, certain businesses must send isolated groups or subcontract the job of development to others at various times. That essentially says you will have to invest twice as much in a single application. Despite the proximity of facilitated endeavors between various parties, synchronization is haphazard, with endeavors to decrease fetched and enhancement time deficient.

In recent times, there were many endeavors to familiarize PhoneGap and HTML5 to make it easier to create applications that are nearly local at all stages by making these local apps using the web views interface [9]. However, the downsides are obvious: this method requires approach to the portable devices which are rich in features, and that could be taken as a workaround rather than a result it could be strived for by the computer program designing team by reusing previously created applications.

2.3 Context Aware Designing of Application

Present-day mobiles are a world separately from traditional computing stages in that their platforms are very active in relationship with the environment, as traditional stages are, with predictable, little changes. In addition, mobile devices are expected to constantly observe the environment, resulting in a mobile application that is largely context-aware like where are you located and at what time etc. [10, 11]. With the

help of applications that are setting conscious, the portable stage contextualizes time, temperature, place, and proximity to deliver ultra-personalized, rich, and energetic content to clients.

Previously, internet applications were able to recognize users based on their dialect, place, and time, and provide them with relevant information. The unused portable stage, on the other hand, allows for a degree of context-aware that has never been possible with traditional approaches to program development, but is possible with the specialist-arranged approach (12). It is critical that we regard context-aware applications in mobile gadgets as a must, so that when computer program architects consider the requirements of a portable stage application, they consider context-aware software which increase the convenience of the devices as shown in Fig 1.

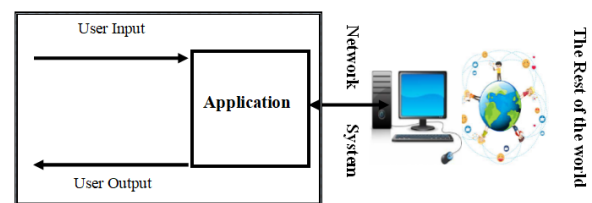


Fig 1: Context Aware System

2.4 Ambiguous and future requirement

Although, designers of mobile applications currently use an adhoc or agile approach, clients are growing tired of unreliable applications and requesting more context-aware applications, and competition among applications is increasing; therefore, a more recognized approach is needed for the development of different types of portable applications. Dexterous construction must also be used for evaluating and indicating the necessity of a variety of applications. Because of their relevant and energetic nature, the demand for applications that adapt naturally is growing, which is especially important where the behavior of application is not able to meet both the functional and non-functional needs. In this case, the application should offer the client with a smaller, low-valuable substance that does not carry such stringent requirements. It is often This is often applicable to those applications which must be carrying the operations even though they are not able to satisfy the result; as a result, they can alter their claim actions by reducing their usefulness rather than being completely non-functional. For example, in area-based applications, the updated and granularity of the material are dependent on various factors such as whether the GPS sensor is turned on or off, or whether the battery is dead. In particular instances, instead of showing an error message the software should display the content in the last accessible field. The necessity of self-adaptation relies on environment is treated in a casual and adhoc manner during the program development stage for flexible applications. In any case, as the need for creating context-aware applications grows, we will need a more systematic approach to computer program development combined with dexterous improvement methods. This will enable application designers to more viably application which can have action in the absence of requirement, allowing them to adjust to function with their prerequisites partially met.

3. PROSPECTIVE RESEARCH AREAS

After discussing the problems and obstacles that the existing software developing stage for mobile devices faces, we will not suggest a few areas where more research and development

could be performed based on the current challenges.

3.1 User Interface for Handicaps

With the advancement of development of mobile application and the growing zone of their application, research into areas of use and accessibility for the impaired or disabled clients would be needed sooner rather than later. According to US Census data, approximately 15% of US people have at least one inability, and are typically not confined to physical or tangible confinements [13]. Nevertheless, we observe hardly any research was done to identify the needs of this particular segment of clients in terms of computer program designing and development of different type of mobile applications for mobile devices.

To accommodate the outwardly impaired, we have an incomplete set of recommendations for progression and modifications. According to these guidelines, a Voice-Over program that could act as help for those users who have poor vision, it could work as screen reader and will not require as additional client [14]. However, we expect to see further research into formative roads for applications aimed at other disabled clients with handling or other physical limitations.

3.2 Product Lines for the development of application

It is critical and fundamental for the computer program designing group of mobile applications to primarily use software designing methods such as SPLE (Software Product Line Engineering) in order to lower the production cost of applications that are essentially comparable and can be apply on all type of platforms. "The capability of re-usability is utilized by SPLE in the development and investigation of mobile applications, and reusability is reinforced by the development of an application set which offers a collection of supervised and shared requirements" [15].

The SPL definition creates the collection of software with several common core requirements, while the rest of the requirements are based on a set of variables, making them distinct [15]. With this strategy, it is possible to reduce the cost and time spent on the development of software, and it "appears to be the most successful approach to intra-organizational software reuse" [16].

Lai & Weiss suggested a two-phase approach to SPLE: first, the organize of space building, in which the needs (variable and common) for the entire item line are defined; second, the organize of application building, in which a particular application are generated within product line. It could be a good strategy for developing mobile apps because it will help engineers focus on the common properties, plans, and requirements for creating mobile apps for a variety of devices (Google, Samsung, HTC, Apple, and so on) and operating systems (Android, iOS, and so on).

If we combined SPLE together with mobile application creation, designers were able to assess the application's specifications from a platform-independent aspect, concentrating on the app's commonalities across its various adaptations. This will also have a global impact by leading the creation of application requirements from the start, rather than creating independent or uncoordinated development and planning groups that may or may not collaborate. Inquiry is needed in determining how SPLE can be made more cost-effective for the advancement of diverse applications and the reduction of repetition of earlier computer program development resources or function.

3.3 Context-Aware Application Development

Context-awareness is a key reason for its ubiquity and a unique feature of mobile applications nowadays. Software designers must use context-aware computer program building methods like AOSE (Agent-Oriented Software Engineering) with the aim of plan scalable applications which are context-aware. AOSE offers a software development methodology, templates, and high levels of reflection for developing free program specialists for use in a MAS. Context-aware is a necessary and vital necessity for a MAS. The MAS operators should be capable of reply and feel the environment to get the desired results (which could be a functional requirement). This is especially true for mobile applications.

There is a need for more research into how AOSE principles can be updated or utilized to produce mobile applications, thereby advancing the design of context-aware software and thereby promoting development of mobile.

3.4 Self-Adaptation of Mobile application

These are heavily reliant on non-functional specifications, and some implementations may be needed to actively self-adapt so that it can provide the less useful functionality. To substantiate this energetic requirement, which can be met by the use of context-aware applications and self-adapting plans, mobile application designers should also adopt approaches for determining prerequisites for self-adjustment frameworks, such as RELAX [17].

RELAX, which is a language that specifies the requirement recommended by Whittle et al., may be a method of expression of actions along with surrounding uncertainty, almost a powerfully adjusting approach [17]. Underneath RELAX, we have a division of necessities into variations (which can be mostly fulfilled) and invariant (which must be basically fulfilled), indicated in a common dialect that is designed on fuzzy logic, and modular administrators which is in according to the fuzzy logic. The RELAX establishes how changes in the environment will affect each version of the requirement, as well as which requirements can be met. Through this method, we can provide requirement catchphrases like 'as many, as near, as early' – as imaginable and inescapably as 'Shall', thus archiving dubious viewpoints and the way computer program is to adjust in the event of dubious conditions during the delivery of little usability.

During the adjustment of a mobile application to RELAX, the programmers must think about how the software will change when it encounters a non-ideal situation or performance. Combining this with the Dexterous method, we can create a prerequisite framework which will be more effective at fulfilling non-functional prerequisites when the context/environment changes [18].

4. CONCLUSION

We have attempted to depict briefly four of the difficulties encountered during the creation of a mobile application in this article. With technological advancements in mobile devices, mobile application development has progressed dramatically in recent years. These enhancements expanded the breadth of applications for mobile devices, and mobile software developers are increasingly taking advantage of these new features, posing new difficulties and opportunities for mobile software development. All-inclusive User Interaction formation, Portability of cross-stage computer software, Context Aware Application, and Handling ambiguous requirement and future requirement. This paper primarily

emphasizes the need of efforts to focus on the development approaches required for the designing of UI, Context aware application and future requirement during its uncertainty. The paper also highlights the need for further research in the areas of UI design approaches in computer programs, constructive reuse of content during the initial developmental stages, the creation of context-aware software, and exploitation dexterous development approaches to cope with ambiguity in system requirements. We also stress the importance of educational activities in the previously mentioned areas, so that the method to make effective communication among engineers who are assigned with task to develop agile software.

5. REFERENCES

- [1] Gartner Group, "Gartner Says More than 1 Billion PCs in Use Worldwide and Headed to 2 Billion Units by 2021," [Online]. Available: <http://www.gartner.com/it/page.jsp?id=703807>. [Accessed:11-Jan-2021].
- [2] A. Oulasvirta, M. Wahlström, and K. Anders Ericsson, "What does it mean to be good at using a mobile device? An investigation of three levels of experience and skill," Mar. 2011.
- [3] A. I. Wasserman, "Software engineering issues for mobile application development," 2010.
- [4] F. Balagtas-Fernandez, J. Forrai, and H. Hussmann, "Evaluation of user interface design and input methods for applications on mobile touch screen devices," 2009.
- [5] G. C. Roman, G. P. Picco, and A. L. Murphy, "Software engineering for mobility: a roadmap," 2000.
- [6] B. Shneiderman, "Designing the user interface," 1987.
- [7] J. Gong and P. Tarasewich, "Guidelines for handheld mobile device interface design," 2004.
- [8] S V N Sreenivasu, Nilamadhab Mishra, Shafiqul Abidin, Nripendra Narayan Das, Dr.P. Siva Kumar Ponnusamy "Implementing the Model for Software Quality Based on Interaction between User and Developer", Journal of Critical Review (JCR), Vol 7, Issue 15, 2020, pp 2512-2518, ISSN: 2394-5125.2009.
- [9] S. Allen, Pro Smartphone Cross-Platform Development: iPhone, Blackberry, Windows Mobile, and Android Development and Distribution, 1st ed. Apress, 2010.
- [10] T. Hofer, W. Schwinger, M. Pichler, G. Leonhartsberger, J. Altmann, and W. Retschitzegger, "Context-awareness on mobile devices - the hydrogen approach," 2003.
- [11] J. Dey, Anind K., Hakkila, "Context-Awareness and Mobile Devices," 2008.
- [12] N. Jennings, "Agent-oriented software engineering," 1999.
- [13] M. Brault, "Disability status and the characteristics of people in group quarters: a brief analysis of disability prevalence among the civilian noninstitutionalized and total populations in the American community survey," US Census Bureau, 2008.
- [14] <http://developer.apple.com/library/ios/#DOCUMENTATION/UserExperience/Conceptual/MobileHIG/Introduction/Introduction.html>.
- [15] D. Weiss and C. Lai, "Software product line engineering: a family-based software engineering process." Addison Wesley, 1999.
- [16] J. Bosch, "From software product lines to software ecosystems," 2009.
- [17] J. Whittle, P. Sawyer, N. Bencomo, B. H. C. Cheng, and J.M. Bruel, "RELAX: a language to address uncertainty in self-adaptive systems requirement," 2010.
- [18] Shafiqul Abidin, Mohd Izhar, Vikas Rao Vadi "5th Generation Wireless Communication Revolution", International Journal of Recent Technology and Engineering, Volume-8, Issues-5, January 2020.(ISSN:2277-3878), Scopus Index).