# The Applicability of Present Estimation Models to the Context of Mobile Applications

Laudson Silva de Souza and Gibeon Soares de Aquino Jr.
*Department of Informatics and Applied Mathematics-DIMAp, Federal University of Rio Grande do Norte, Natal, Brazil*

Keywords: Software Engineering, Software Quality, Estimating Software, Systematic Review, Mobile Applications, Mobile Computing.

Abstract: The growing use of mobile technologies has shown different ways to access information and interact with other computer systems. Thus, the traditional information systems are undergoing a process of adaptation to this new computing environment. Thereafter, there is a need to reassess the current knowledge on the planning and development of systems in this new environment. One area in particular that demand such adaptation is the estimation software. The estimation processes, in general, are based on characteristics of the systems, trying to quantify the complexity of implementing them. Hence, the main objective of this paper is to present a proposal for an estimation model for mobile applications, and to debate about the applicability of the traditional estimation models on this environment. Throughout the paper we analyze existing methods of estimates, identify specific features of systems for mobile devices and finally an adaptation to be proposed for this area of an existing estimation method.

## 1 INTRODUCTION

Computing is becoming increasingly present in people's lives and currently in a much more intense and accelerated way due to the rise of the use of mobile technologies in the world, such as mobile phones, smartphones and tablets, all connected to mobile networks, which are increasingly more present in many places and with better speeds. We are facing a new technological scenario that is changing old habits and creating new ways for the society to access information and interact with computer systems (Naismith et al., 2004), (Macario et al., 2009) and (Liu et al., 2003).

The fact is that this new technological scenario that is emerging with new requirements and restrictions requires a reevaluation of current knowledge about the processes of planning and building software systems. These new systems have different characteristics and, therefore, an area in particular that demands such adaptation is software estimation. The estimation processes, in general, are based on characteristics of the systems, trying to quantify the complexity of implementing them. For this reason, it is important to analyze the methods currently proposed for software projects estimation and evaluate their applicability to this new context of mobile computing.

Hence, the main objective of this paper is to present a proposal for an estimation model for mobile applications, as well as discuss the applicability of traditional models used in estimation of information systems for the purpose of the development of systems in the context of mobile computing. In this work, the main estimation methods that exist now will be analyzed, the specific characteristics of mobile systems will be identified and an adaptation of a estimation method that exists in this context will be proposed.

## 2 MAIN ESTIMATION METHODS

In order to identify how the traditional estimation methods could address the characteristics of the systems, a literature review on the main estimation methods was performed. The methods recognized by ISO identified in the survey can be seen in Table 1. All methods with their descriptions identified in the survey can be accessed at http://www.laudson.com/methods.pdf.

Table 1 displays in chronological order the estimation methods recognized by ISO, showing the year of creation, the name of the method and the author of it. At first glance, one realizes that the main existing methods were not designed to consider the re-

Silva de Souza L. and Soares de Aquino Jr. G..
The Applicability of Present Estimation Models to the Context of Mobile Applications.
DOI: 10.5220/0004973002490254
In *Proceedings of the 9th International Conference on Evaluation of Novel Approaches to Software Engineering* (ENASE-2014), pages 249-254
ISBN: 978-989-758-030-7

Table 1: Estimation Methods Recognized by ISO.

| Year | Method | Author |
|---|---|---|
| 1979 | Function Point Analysis (FPA) | Albrecht (Oligny et al., 1999) |
| 1988 | Mark II FPA | Charles Symons (Symons, 2001) |
| 1990 | Netherlands Software Metrics Users Association (NESMA) FPA | The Netherlands Software Metrics Users Association (Engelhart et al., 2001) |
| 1999 | Common Software Measurement International Consortium (COSMIC) FFP | Common Software Measurement International Consortium (COSMIC) (Consortium et al., 2007) |
| 2004 | Finnish Software Metrics Association FSM | The Finnish Software Metrics Association (FiSMA) (Forselius, 2004) |

quirements of mobile applications. Indeed, the very creation of most of them precedes the emergence of mobile devices as we know today. This suggests that the use of these methods to estimate the effort of the development of projects involving systems or applications for mobile devices would cause a possible failure to quantify the complexity of some features and, therefore, would not produce adequate estimates.

# 3 CHARACTERISTICS OF MOBILE APPLICATIONS

In order to identify characteristics that are inherent to systems and mobile applications, a surveying of the characteristics of these types of software was accomplished through a systematic review. Conducting a systematic review is relevant because most searches begin with some kind of review of the literature, and a systematic review summarizes the existing work fairly, without inclinations. So the surveys were conducted according to a predefined search strategy, in which the search strategy should allow the integrity of the research to be evaluated. The planning and accomplishment of the methodology discussed were directed by *Procedures for Performing Systematic Reviews* (Kitchenham, 2004).

## 3.1 Systematic Review

In the context of research questions, the following research question was formulated: “What are the characteristics of Mobile Applications?”, based on the issue about the proposed study.

Procedures for The Evaluation of the Articles: the articles will be analyzed considering its relation with the issues addressed in the research questions, inclusion criteria and exclusion criteria, and their respective situation will be assigned with either ”Accepted” or ”Rejected”. The evaluation will follow the following procedure: read the title and abstract and, should it be related with the research question, also read the whole article. The implementation of the systematic review was performed almost in line with its planning, except for the need to adjust the syntax of the proposed search string due to the particularities of the research bases. 234 articles were analyzed, of which 40 were selected and considered “Accepted” according to the inclusion criteria; 194 were considered “Rejected” according to the exclusion criteria. The list with all the articles can be accessed at the following address: http://http://www.laudson.com/sr-articles.pdf.

Given the results extracted from the systematic review, it’s is possible to identify 29 kinds of characteristics in 100% of the articles evaluated and considered accepted in accordance with the inclusion criteria. However some of these are a mixture of characteristics of mobile devices and characteristics of mobile applications, such as the characteristic called “Limited Energy”, which is a characteristic of the device and not the application, however the articles that mention this type of characteristic emphasize that in the development of a mobile application, this “limitation” must be taken into account since all the mobile devices are powered by batteries, which have a limited life, depending completely on what the user operates daily. Applications requiring more hardware or software resources will consume more energy. The 23 types of characteristics mentioned the most in the selected articles can be observed following. There is a description of each characteristic identified in the review:

○ Limited energy (Sohn et al., 2005); Small screen (Sohn et al., 2005); Limited performance (Mukhtar et al., 2008); Bandwidth (Mukhtar et al., 2008); Change of context (Mukhtar et al., 2008); Reduced memory (Sohn et al., 2005); Connectivity (Feng, 2009); Interactivity (Mukhtar et al., 2008); Storage (Mukhtar et al., 2008); Software portability (Mukhtar et al., 2008); Hardware portability (Mukhtar et al., 2008); Usability (Feng, 2009); 24/7 availability

(Feng, 2009); Security (Feng, 2009); Reliability (Kumar Maji et al., 2010); Efficiency (Al-Jaroodi et al., 2009); Native vs. Web Mobile (Feng, 2009); Interoperability (Mukhtar et al., 2008); Response time (Hameed et al., 2010); Privacy (Feng, 2009); Short term activities (Alekhya Mandadi, 2009); Data integrity (Hameed et al., 2010); Key characteristics (Alekhya Mandadi, 2009); Complex integration of real-time tasks (Hayenga et al., 2008); Constant interruption of activities (Alekhya Mandadi, 2009); Functional area (Giessmann et al., 2012); Price (Giessmann et al., 2012); Target audience (Giessmann et al., 2012); Provider type (Giessmann et al., 2012).

# 4 CHARACTERISTICS OF MOBILE APPLICATIONS SURVEY

With the conclusion of the systematic review, a survey was carried out among experts in mobile development with the purpose of ratifying the characteristics previously raised and to prove their respective influence on mobile development. The disclosure of the survey was conducted in more than 70 locations, among them universities and businesses, through e-mails, study groups and social groups. In general, of all 117 feedbacks received through the survey, 100% of the experts confirmed the characteristics; among them, an average of 72% indicated a greater effort and complexity regarding the characteristics during development, an average of 12% indicated less effort and complexity and, finally, an average of 16% indicated they did not perceive any difference in mobile development, even though they confirmed the presence of the characteristics.

# 5 PROBLEM ADDRESSED

As noted in Section II, there is no estimation method developed for mobile applications projects. Moreover, some of the characteristics elicited in Section III aggravate the complexity and, thereafter, the effort in the development of mobile applications. From the analysis that follows, with the characteristics of applications on mobile devices elicited in Section III, it is clear that they are different from the characteristics of traditional systems and directly influence its development. A clear example, which is different from the information or desktop systems, is the characteristic that the mobile devices have “Limited Energy”. As mobile devices are powered by battery, which have a limited lifetime period, the applications must be programmed to require the minimal amount of hardware resources possible, since the more resources consumed, the greater amount of energy expended. This characteristic makes it necessary for the solution project to address this concern, generating a higher complexity of development and, thereafter, a greater effort and cost.

Another specific characteristic of this context is the “Graphical Interface”. Due to the reduced screen size, the interface design is limited. Therefore, a greater complexity and, thereafter, a larger effort is required in the development of the graphical interface. Another characteristic related to the screen is the “Input Interface”, which defines how the user will interact with the application, in other words, if the user will interact via keypad, stylus, touch screen or voice and image recognition. The latter makes the task of developing applications that offers all these interaction options more complex, thus requiring a bigger effort.

Regarding connectivity, the characteristic “Bandwidth” was identified, wherein a mobile application might have the maximum band at times and the minimum in other moments. Some types of applications need to realize this and act differently in each situation. Another related feature is the “Connectivity Type”. Mobile applications can be developed to support different types of connectivity such as 3G, bluetooth, infrared, Wi-Fi, Wireless, NFC and others. In addition, a single application can support multiple types of connectivity simultaneously. These behaviors directly affect the complexity of the software and therefore require a larger development effort.

The “Change in Context” is also another characteristic inherent in mobile applications, which should take into account not only the data entries explicitly provided by users, but also the implicit entries concerning the physical and computational context of the users and the environments that surround them. In addition, the “Constant Interruption of Activities” is a much more common characteristic in this context, as well as the need for some applications to be developed to work offline and therefore be able to synchronize. Mobile applications should be prepared for different scenarios because the activities are interrupted constantly. Receiving a call, lack of connection and low battery are examples of such interruptions, which makes the applications become much more complex.

Despite the advances related to the computational ability of these devices, their hardware must still be considered as limited, especially when compared to desktops and servers. Two characteristics related to this issue are “Limited Performance” and “Reduced

Memory". Besides these, a characteristic inherent to the use of mobile devices is the "Response Time", that is directly related to the power of "Processing". Mobile applications must be initialized and finalized immediately, in other words, any development should be focused in the time variable. These characteristics require the applications to be developed with a possible resource optimization for a better efficiency and response time, requiring more effort.

The "Portability" is also a required characteristic of these applications. It can be divided into two characteristics: the "Hardware Portability" and the "Software Portability". Regarding the first one, nowadays there is a large number of different devices with different capabilities and resources. A mobile application should be able to run on the largest number of devices possible. This requires an increased effort in the development. Moreover, a greater effort in testing this kind of portability is required. Regarding "Software Portability", it is necessary to develop specific applications for each existing platform should the application be native. With this, more effort is required for replications of the same software product, including the tests.

Finally, mobile applications can be separated into two types: Native or Web Mobile. The first one has higher performance and easiness in accessing the hardware, while the second has lower performance since it is web, but it is easier to achieve portability. In addition, there are some applications that are considered hybrids. Depending on the type of application, the issues that must be considered and the complexity can be different, requiring different development efforts.

From the survey of the most popular estimation methods cited in Section III, it was found that these characteristics are not covered by the current estimation methods for two explicit reasons: first, none of the existing methods was designed to perform project estimation in mobile applications development; and second, all the characteristics discussed in this section are exclusive to mobile applications, with direct interference in their development, thereby generating a greater complexity and, thereafter, a greater effort. However, to consider any of the existing estimation methods to apply to the process of development of mobile applications is to assume that this kind of development is no different than the project of developing desktop applications, in other words, an eminent risk is assumed.

# 6 PROPOSAL: ESTIMATION IN MOBILE APPLICATION DEVELOPMENT PROJECT

The approached proposed is an adaptation of an existing method, based exclusively on methods recognized as international standards by ISO. Among the most popular estimation methods mentioned in Section III, the method used to base the proposal below on is known as "*Finnish Software Metrics Association* (FISMA)". The model is one of the five methods for measuring software that complies with the ISO/IEC 14143-1 standard, is accepted as an international standard for software measuring (Forselius, 2004) and nowadays over 750 software projects are completed being estimated by FISMA. However, the difference between this and other methods that are in accordance with the above standard, which are the *Common Software Measurement International Consortium Function Points* (COSMIC FP) (Consortium et al., 2007), the *International Function Point Users Group* (IFPUG) FPA (Oligny et al., 1999), *MarkII FPA* (Symons, 2001) and the *Netherlands Software Metrics Association* (NESMA) WSF (Gencel et al., ), is that the method used is based in functionality but is service-oriented. It also proposes in its definition that it can be applied to all types of software, but this statement is lightly wrong since in its application, the method does not take into account the characteristics elicited in Section IV.

Overall, the FISMA method proposes that all services provided by the application are identified. It previously defines some services, among which stands out the user's interactive navigation, consulting services, user input interactive services, interface services for other applications, data storage services, algorithmic services and handling services. Finally, after identifying all the services, the size of each service is calculated using the same method and thus obtaining a total functional size of the application by adding the size of each service found (FiSMA, 2004).

## 6.1 Applying the Chosen Model

The FiSMA method can be applied manually or with the aid of the Experience Service[1] tool, which was the case, provided by FiSMA itself through contact made with senior consultant Pekka Forselius and with the chairman of the board Hannu Lappalainen.

When using the tool, it is necessary to perform all the steps of the previous subsection to obtain the functional size. Figure 1 shows the final report after the

[1] http://www.experiencesaas.com/

implementation of the FiSMA on a real system, the Management of Academic Activities Integrated System (Sigaa) in its Mobile version, developed by the Superintendence of Computing (SINFO) of the Federal University of Rio Grande do Norte (UFRN).

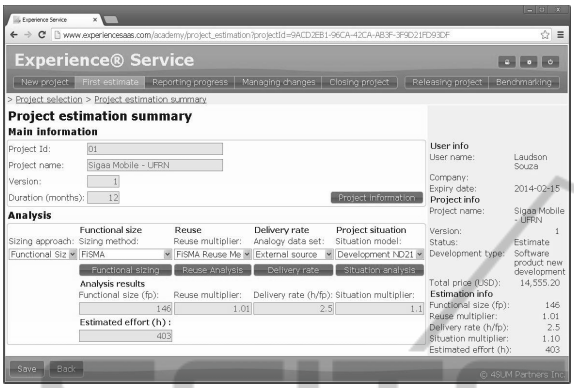

Figure 1: Final Report of FiSMA applied to Sigaa Mobile.

After the application of FiSMA, the functional size of the software is obtained and from this it is possible to find the effort using the formula: Estimated effort (h) = size (fp) x reuse x rate of delivery (h/fp) x project status; the latter is related to productivity factors that are taken into account for the calculation of the effort. However, of the factors predefined by the FiSMA regarding the product, only 6 (six) are proposed, in which the basic idea of the evaluation is that "the better the circumstances of the project, the more positive the assessment". The weighting goes from - - to + +, as follows: **Caption:** ○ (+ +) = [1.10] Excellent situation, much better circumstances than in the average case; ○ (+) = [1.05] Good situation, better circumstances than in the average case; ○ (+ / -) = [1.0] Normal situation; ○ (-) = [0.95] Bad situation, worse circumstances than in the average case; ○ (- -) = [0.90] Very bad situation, much worse circumstances than in the average case.

Among the productivity factors mentioned above, only the "Portability Requirement" factor fits in harmony with the "Portability" characteristic regarding both hardware and software. However, none of the other factors discusses the characteristics of mobile application, in other words, after obtaining the functional size of the software and applying the productivity factors related to the product to estimate the effort, this estimate ignores all of the characteristics of mobile applications, judging that the estimate of traditional information systems is equal to the mobile application. However, with the proposal of the creation of new productivity factors, which would be the specific characteristics of mobile applications, this problem will be solved, as presented below.

Performance Factor: ○ (-) The application should be concerned with the optimization of resources for a better efficiency and response time. ○ (+/-) Resource optimization for better efficiency and response time may or may not exist. ○ (+) Resource optimization for better efficiency and response time should not be taken into consideration.

Power Factor: ○ (-) The application should be concerned with the optimization of resources for a lower battery consumption. ○ (+/-) Resource optimization for lower battery consumption may or may not exist. ○ (+) Resource optimization for a lower battery consumption should not be taken into consideration.

Band Factor: ○ (-) The application shall require the maximum bandwidth. ○ (+/-) The application shall require reasonable bandwidth. ○ (+) The application shall require a minimum bandwidth.

Connectivity Factor: ○ (-) The application must have the maximum willingness to use connections such as 3G, Wi-fi, Wireless, Bluetooth, Infrared and others. ○ (+/-) The application must have reasonable predisposition to use connections such as 3G, Wi-Fi and Wireless. ○ (+) The application must have only a predisposition to use connections, which can be: 3G, Wi-fi, Wireless, Bluetooth, Infrared or others.

Context Factor: ○ (-) The application should work offline and synchronize. ○ (+/-) The application should work offline and it is not necessary to synchronize. ○ (+) The application should not work offline.

Graphic Interface Factor: ○ (-) The application has limitations due to the screen size because it will be mainly used by cell phone users. ○ (+/-) The application has reasonable limitation due to the screen size because it will be used both by cell phone and tablet users. ○ (+) The application has little limitation due to the screen size because it will be mainly used by tablet users.

Input Interface Factor: ○ (-) The application must have input interfaces for touch screen, voice, video, keyboard and others. ○ (+/-) The application must have standard input interfaces for keyboard. ○ (+) The application must have any one of the types of interfaces, such as: touch screen, voice, video, keyboard or others.

The proposed factors take into account the same weighting proposed by FiSMA, but only ranging from - to +, in other words: ○ (+) = [1.05] Good situation, better circumstances than in the average case; ○ (+ / -) = [1.0] Normal Situation; ○ (-) = [0.95] Bad situation, worse circumstances than in the average case. The functional size remains the same, thus affecting only the formula used to obtain the effort, which will now consider in its "project situation" variable the new productivity factors specific for mobile applications.

# 7 CONCLUSIONS

Given the results presented, based on the literature review of estimation methods and on the systematic review of the characteristics of mobile applications, it was observed that this sub-area of software engineering still falls short. Basically, it's risky to use any existing estimation method in development projects for mobile applications, as much as there are some models already widespread in industry, such as the Function Point Analysis, the Mark II and the COSMIC-FFP, which are even approved by ISO as international standards. They all fall short by not taking into account the particularities of mobile applications, which makes the method partially ineffective in this situation.

The validation shall be as follows, will be raised the total effort expended in developing the Sigaa Mobile project. After the method is applied to estimate FISMA, in his original proposal thus obtaining an estimate of effort. Then the proposed adjustment will be applied also generating an effort estimate finally the comparative analysis between the three estimates generated will be performed to determine which proposal is closer to the actual effort spent.

Based on this study, it is concluded that the proposal presented in this work is entirely appropriate and viable and that this proposal should take into account all the peculiarities of such applications, finally creating a belief that there actually are considerable differences in the development project for mobile applications.

# REFERENCES


Al-Jaroodi, J., Al-Dhaheri, A., Al-Abdouli, F., and Mohamed, N. (2009). A survey of security middleware for pervasive and ubiquitous systems. In *Network-Based Information Systems, 2009. NBIS '09. International Conference on*, pages 188–193. IEEE.

Alekhya Mandadi, Deepti Mudegowder, I. R. D. E. A. G. (2009). Mobile applications: Characteristics & group project summary. *Mobile Application Development*. Google.

Consortium, C.-C. S. M. I. et al. (2007). The cosmic functional size measurement method-version 3.0 measurement manual (the cosmic implementation guide for iso/iec 19761: 2003).

Engelhart, J., Langbroek, P., et al. (2001). *Function Point Analysis (FPA) for Software Enhancement*. NESMA.

Feng, H. (2009). A literature analysis on the adoption of mobile commerce. In *Grey Systems and Intelligent Services, 2009. GSIS 2009. IEEE International Conference on*, pages 1353–1358. IEEE.

FiSMA, F. S. M. A. (2004). Fisma functional size measurement method version 1-1.

Forselius, P. (2004). Finnish software measurement association (fisma), fsm working group: Fisma functional size measurement method v. 1.1.

Gencel, C., Heldal, R., and Lind, K. On the conversion between the sizes of software products in the life cycle.

Giessmann, A., Stanoevska-Slabeva, K., and de Visser, B. (2012). Mobile enterprise applications–current state and future directions. In *System Science (HICSS), 2012 45th Hawaii International Conference on*, pages 1363–1372. Google.

Hameed, K. et al. (2010). Mobile applications and systems. Google.

Hayenga, M., Sudanthi, C., Ghosh, M., Ramrakhyani, P., and Paver, N. (2008). Accurate system-level performance modeling and workload characterization for mobile internet devices. In *Proceedings of the 9th workshop on MEmory performance: DEaling with Applications, systems and architecture*, MEDEA '08, pages 54–60, New York, NY, USA. ACM. ACM.

Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33:2004.

Kumar Maji, A., Hao, K., Sultana, S., and Bagchi, S. (2010). Characterizing failures in mobile oses: A case study with android and symbian. In *Software Reliability Engineering (ISSRE), 2010 IEEE 21st International Symposium on*, pages 249–258. IEEE.

Liu, T., Wang, H., Liang, J., Chan, T.-W., Ko, H., and Yang, J. (2003). Wireless and mobile technologies to enhance teaching and learning. *Journal of Computer Assisted Learning*, 19(3):371–382.

Macario, G., Torchiano, M., and Violante, M. (2009). An in-vehicle infotainment software architecture based on google android. In *Industrial Embedded Systems, 2009. SIES '09. IEEE International Symposium on*, pages 257–260.

Mukhtar, H., Belaïd, D., and Bernard, G. (2008). A model for resource specification in mobile services. In *Proceedings of the 3rd international workshop on Services integration in pervasive environments*, SIPE '08, pages 37–42, New York, NY, USA. ACM. ACM.

Naismith, L., Sharples, M., Vavoula, G., Lonsdale, P., et al. (2004). Literature review in mobile technologies and learning.

Oligny, S., Desharnais, J.-M., and Abran, A. (1999). A method for measuring the functional size of embedded software. In *3rd International Conference on Industrial Automation*, pages 7–9.

Sohn, J.-H., Woo, J.-H., Lee, M.-W., Kim, H.-J., Woo, R., and Yoo, H.-J. (2005). A 50 mvertices/s graphics processor with fixed-point programmable vertex shader for mobile applications. In *Solid-State Circuits Conference, 2005. Digest of Technical Papers. ISSCC. 2005 IEEE International*, pages 192–592 Vol. 1. Google.

Symons, C. (2001). Come back function point analysis (modernized)–all is forgiven!). In *Proc. of the 4th European Conference on Software Measurement and ICT Control, FESMA-DASMA*, pages 413–426.