Mobile Applications: The Paradox of Software Estimation

Laudson Silva de Souza
Department of Informatics and Applied Mathematics
Federal University of Rio Grande do Norte
Natal, Brazil 59078-970
Email: laudyson@gmail.com

Gibeon Soares de Aquino Jr.
Department of Informatics and Applied Mathematics
Federal University of Rio Grande do Norte
Natal, Brazil 59078-970
Email: gibeon@dimap.ufrn.br

Abstract—There is a global trend towards the increase of the number of users connected to the network via mobile devices which, consequently, will create an increasing demand for information, applications and content for such equipments. New ways to use existing aplicações are emerging. In particular, systems that were once accessed via web interfaces through personal computers physically located in offices, universities or homes are providing new ways to access from mobile devices which, in turn, have different requirements and capabilities than the personal computers. Hence, the main objective of this paper is to present an estimation model for mobile applications.

Keywords—Quality, Estimating, Software Engineering, Mobile Applications

I. INTRODUCTION

The remarkable growth in the use of mobile devices in several areas, such as health, work, education and public safety. Come along with the accession of all kinds of mobile applications, but those interested in the area awakens a huge discussion on the approach to estimating software, for this private context. For although there was no specific method to handle the particularities of this area. And the software to be developed for mobile applications, are estimated with traditional methods of estimates, ie, facing desktop applications. This new technological scenario is changing old habits and creating new ways of society to access information and interact with computer systems [1], [2] e [3].

According to Gartner, 1.75 billion people own mobile phones with advanced capabilities and foresees further growth in the use of this technology in the coming years [4]. The ITU \(^1\) estimates that there are more than six (6) billion mobile subscribers worldwide. There is a worldwide trend towards increasing the number of users connected to the network via mobile devices, which consequently produce an increasing demand for applications and content for such equipment. With ways of using information systems emerging, such as the access that were previously performed through Web interfaces for personal computers in offices, universities or homes, are providing new ways to access from mobile devices, which turn have different requirements and capabilities of personal computers.

There is a global trend towards the increase of the number of users connected to the network via mobile devices which, consequently, will create an increasing demand for information, applications and content for such equipments. New ways to use existing aplicações are emerging. In particular, systems that were once accessed via web interfaces through personal computers physically located in offices, universities or homes are providing new ways to access from mobile devices which, in turn, have different requirements and capabilities than the personal computers. Hence, the main objective of this paper is to present an estimation model for mobile applications, denominado “MEstiAM” (Estimation Model for Mobile Applications) proposed in [5].

II. PRESENTING THE MODEL

According to the proposed [5], which conducts a literature review covering the main estimation methods, but also conducts a systematic review of the characteristics of mobile applications and conducts a survey to ratify the specific characteristics of context finally proposes the method previously cited as a specific method of estimation for the design of mobile application development.

A. Approaching the Model

The MEstiAM model is totally based on FISMA method, which in its original usage proposes a structure of seven classes of the Base Functional Component or BFC (Base Functional Component) type, which is defined as a basic component of functional requirement. The seven classes used to account for the services during the application of the method are [6]: interactive navigation of the end user and query services (q); interactive input services from end users (i); non-interactive outbound services for the end user (o); interface services for another application (t); interface services for other applications (l); data storage services (d) and algorithmic manipulation services (a).

The identification for each class name BFC previously mentioned, with a letter in parenthesis, is used to facilitate the application of the method during the counting process, because each of the seven classes BFCs are composed of other BFC classes which, at the time of calculating, these BFCs “daughter” classes are identified by the letter of their BFC “mother” class followed by a numeral, as can be seen in Figure 1.

The unit of measurement is the point of function with the letter “F” added to its nomenclature to identify the “FiSMA”, \(^1\)International Telecommunication Union
resulting in FfP (FiSMA Function Point) or Ffsu (FiSMA functional size unit). The measurement process generally consists of measuring the services and end-user interface and the services considered indirect [6], as can be seen in Figure 2.

Figure 2 shows the process of measuring the base model, in which it defines each step and sum of each BFC class of the model. Briefly, the process of counting should be done as follows. Identify:

- How many types of BFCs does the software have?
- Which are they? (identify all)
- What are they? (provide details of each BFC identified).

After doing this, it is necessary to add each BFC root using the formulas pre-defined by the method and their assignments. Finally, the formula of the final result of the sum is the general sum of all the BFCs classes.

B. Applying the Model

The FiSMA method can be applied manually or with the aid of the Experience Service 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 3 shows the final report after the 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).

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:

<table>
<thead>
<tr>
<th>Weighting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(+ +) = [1.10]</td>
<td>Excellent situation, much better circumstances than in the average case;</td>
</tr>
<tr>
<td>(+) = [1.05]</td>
<td>Good situation, better circumstances than in the average case;</td>
</tr>
<tr>
<td>(+ / -) = [1.0]</td>
<td>Normal situation;</td>
</tr>
<tr>
<td>(-) = [0.95]</td>
<td>Bad situation, worse circumstances than in the average case;</td>
</tr>
</tbody>
</table>

---

Fig. 3. Final Report of FiSMA applied to Sigaa Mobile.

http://www.experiencesaas.com/
• (- -) = [0.90] Very bad situation, much worse circumstances than in the average case.

Productivity factors:

• Functionality requirements - compatibility with the needs of the end user, the complexity of the requirements.
  ◦ (- -) Complex and critical application area (thousands of FPs), multiple users and multicultural system.
  ◦ (-) Interoperable application area with some complex characteristics, requiring special understanding from users and developers.
  ◦ (+/-) Partly automated, integrated application area and a medium size application (between 600 and 1000 FPs) with standard security requirements.
  ◦ (+) Application area mostly automated and application with less than 5 interfaces with other systems; there are specific security requirements.
  ◦ (+++) Very mature application area, simple and easy, a small stand-alone application (less than 200 FPs) for a small group of users.

• Reliability requirements: maturity, tolerance to faults and recovery for different types of use cases.
  ◦ (- -) Malfunctions may put in danger human lives and cause significant economic or environmental losses.
  ◦ (-) The software is part of a large real-time system where all the failures of operation will cause problems to many other applications.
  ◦ (+/-) Not more than 2 hours of downtime is acceptable, but the system recovery routines are appropriate.
  ◦ (+) Need for non-continuous operation, but daily.
  ◦ (++) Need for periodic operation. Pausing for a few days will not cause any damage to the organization.

• Usability requirements: understandability and easiness to learn the user interface and workflow logic.
  ◦ (-) A large number of different types of end users around the world.
  ◦ (-) 2 or 3 different types of users with different skills.
  ◦ (+/-) A large number of end users with equal abilities.
  ◦ (+) No more than tens or hundreds of homogeneous users in perhaps more than one location.
  ◦ (+++) Only a few users, all located on one site.

• Efficiency requirements: effective use of resources and adequate performance in each use case and under a reasonable workload.
  ◦ (-) Complex database with millions of data records and transactions per day, thousands of simultaneous end users.
  ◦ (-) Large database, hundreds of simultaneous end users, critical response most of the time.
  ◦ (+/-) Large database, less than millions of data records and less than hundreds of simultaneous end users.
  ◦ (+) Medium database in volume and structure, simple and predictable data requests from some simultaneous end users.
  ◦ (+++) Simple and small database without simultaneous end users or complex data requests.

• Maintainability requirements: lifetime of the application, criticality of fault diagnosis and test performance.
  ◦ (- -) Very large strategic software (over 20 years of lifetime) in a volatile area of business, with frequent changes in laws, regulations and business rules.
  ◦ (-) Large software (10-20 years of lifetime), and frequent changes in laws, regulations and business rules.
  ◦ (+/-) Medium size software (5-10 years of lifetime), monthly changes in laws, regulations and business rules.
  ◦ (+) Small software, rarely changes (2 to 5 years of lifetime).
  ◦ (+++) Temporary software (less than 2 years of lifetime), without modifications.

• Portability requirements: adaptability and instability to different environments, to the architecture and to structural components.
  ◦ (- -) Software users are located in many types of organizations, with various platforms (hardware, browsers, operating systems, middleware, protocols, etc), various versions and various update frequencies.
  ◦ (-) The software must operate on some different platforms (hardware, browsers, operating systems, middleware, protocols, etc) and in various versions of each of them.
  ◦ (+/-) Each version of the software must run on multiple versions of a given platform (hardware, browser, operating system, middleware, protocols, etc), and the frequencies of update of the users are quite predictable.
  ◦ (+) The software must run on a given platform (hardware, browser, operating system, middleware, protocols, etc).

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

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

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

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

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

• Input Interface Factor:
  o (-) The application must have input interfaces for touch screen, voice, video, keyboard, and others.
  o (+/-) The application must have standard input interfaces for keyboard.
  o (+) The application must have 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.

The validation process was as follows, was raised the total effort expended in developing the Sigaa Mobile project, ie, we obtained the actual effort. After we applied the method of estimation FISMA, in its original proposal thus obtaining an estimate of effort. Then we applied the method MEStiAM also generating an effort estimate finally the comparative analysis between the three estimates generated was performed to verify which method is closer to the actual effort spent. As can be seen in Table I.

<table>
<thead>
<tr>
<th>TABLE I. ANALYSIS OF ESTIMATES OF SIGAA MOBILE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real Effort Spent</td>
</tr>
<tr>
<td>--------------------</td>
</tr>
<tr>
<td>860 h</td>
</tr>
</tbody>
</table>

As can be seen in Table I, the proposed method, MEStiAM, which is closest to the actual effort spent. You FISMA model, it was very much desired for the new model and the actual effort expended.

III. CONCLUSION

Based on these results, it was evident the difference between a specific method for this context MEStiAM and other methods. Since they fall short by not taking into account the particularities of mobile applications, making it partially ineffective method in this situation.

Finally, it is concluded that it is entirely appropriate and feasible to use the method MEStiAM where it should take into account all peculiarities of such applications. Putting up finally a division that actually there are considerable differences in the design development of mobile applications.

REFERENCES