# Influencing Factors of Mobile Applications' Quality Metrics

Lecturer Paul POCATILU, PhD
Economic Informatics Department, Academy of Economic Studies, Bucharest

*Mobile applications are used in various domains and the development of them has an ascending trend. Mobile applications' quality is a very important aspect that has to be considered carefully. In this paper are presented some influencing factors of mobile applications' quality metrics, having in mind that the quality metrics are quite similar to the classical applications.*
***Keywords**: mobile applications, software quality, metrics.*

## Introduction

Software metrics are very important in the software development process, they allow the quantification of the software, software related deliverables, and the software processes.

The software metrics are defined in [SOMM01] as *any type of measurement which relates to a software system, process or related documentation*.

Software metrics are very important in the software development process, they allow the quantification of the software, software related deliverables, and the software processes.

The existence of software metrics in a software company is a great advantage. The software metrics could be used to improve the software development process, and to predict the software projects' evolution.

Software development cycle, environment, and hardware issues are the main causes for unreliable software.

Software development cycle issues are the most important cause of the unreliability. Poor specifications, inadequate activities of design, implementation, testing and debugging can lead to unreliable software applications.

Environment issues are generated by software incompatibilities between the real environment where the application is running, and the environment that was described by specifications.

Hardware issues can be the result of incompatibilities between different hardware components of the computer on which the application is used.

There are defined many quality metrics include. Among them, there are:

- software size
- manpower effort
- complexity
- number of system classes used
- number of inherited classes
- number of comments
- number of documentation pages

There are several software quality models, (like McCall's, Boehm's) where are described software quality characteristics and the relationships between them

The metrics depend on different characteristics of mobile application environment. Such factors that influence mobile applications metrics are related to operating system (portability), device specific (memory, processor speed, screen size, keyboard, touch screen), network access, and application type (Web based, desktop, client-server).

**Device dependent related factors**

Based on the mobile devices characteristics, the mobile applications have some limits, compared with desktop applications:

- the screen size and resolution leads to a limited interface components;
- the applications size is smaller;
- the complexity of mobile applications is reduced;
- low transfer speed.

Device specific characteristics are: memory, processor speed, screen size, keyboard, touchscreen. Table 1 presents the common mobile devices characteristics.

The biggest issue for these applications is the display's size that allows showing small pieces of data. Thus, the user interface has to be designed for these devices, graphical elements to fit in the display, to keep the infor-

mation together, not to get the user from the context. Another important issue is the limited memory that affects code and data storage.

Regarding the devices' operating system, each operating system (Windows Mobile, Symbian, Palm OS, Linux) has specific APIs, that have different degree of complexity and architectures and are more or less well documented. There are also provided development frameworks and integrated development environments.

**Table 1** Common mobile devices characteristics

| Characteristic / Device | Screen size | Processing power | Memory (RAM/External) | Connectivity |
|---|---|---|---|---|
| Mobile phone | 1"-2.5" | Minimal | 1-64 MB | WAP, GPRS, UMTS, Bluetooth |
| Smartphone | 2.5"-4", 160x160 – 240 x 320 | 144-200 MHz | 32-128 MB/mSD, MMC | GPRS, CDMA2000, WiFi (802.11b), Bluetooth |
| PDA | 2.5"-4", 160x160 – 320x320 | 126-400 Mhz | 16-128 MB/ SD, CFII | WiFi (802.11b), Bluetooth |

In order to increase the productivity, libraries of classes that wrap system's APIs were developed. For each library there is a specific run-time environment.

The size, complexity and productivity are influenced by the application's operating system. Using J2ME technology there is a high degree of portability between operating systems, but here are device specific influences.

The use of native APIs to write applications requires more effort, and the size of application (expresses as KLOC) is higher than using classes libraries.

**Software development related factors**

The development process is made using software emulators of devices, after that the application being deployed on the device. Not all functions can be tested on the emulators so there is an additional effort in testing the application.

Mobile applications are used on various domains and their use is growing each day. The mobile applications are divided in stand-alone or desktop applications and distributed applications. Stand-alone mobile applications are designed to perform specific tasks without the need of a network connection. Mostly mobile applications made for PDAs are such examples of stand-alone applications.

The application type has an impact of the computed metrics. The application type influences the size, and the complexity. As it can be seen from the table 2 [POCA05], mobile applications that require network access and those that use databases usually have a higher complexity. That result is based on assumption that the size of program will be higher, more classes will be used, and there is a need a specific knowledge.

Web-based mobile applications are based on a thin client, like a Web browser, that loads Web pages developed specially for these devices. Such navigators are Pocket Internet Explorer (for Windows Mobile-based devices), Openwave, Go.Web, Palm Web Clipping. Mobile devices navigators use different markup languages. The Web server – through a special gateway – has to convert the HTML content to specific mobile device navigator content. The WML pages contains simple graphical elements, like text, text boxes, buttons, links and black and white pictures.

**Table 2** Comparison of mobile applications types

| Application Type | User Interface | Memory | Processing power | Complexity |
|---|---|---|---|---|
| Network access | Limited | High | Medium/High | High |
| Stand-alone | Limited | Medium/High | Medium/High | Medium/High |
| Web-based | Web-based | Medium | Low/Medium | Low |
| Database access | Limited | High | Medium/High | High |

There is an additional effort to test the mobile applications on different devices, with different characteristics. The software reliability is a very important software quality characteristic. Software reliability is the probability that a software application will work without failure as is provided in specifications. The reliability of a program is

computed using the following metric:

$$R = \frac{N_S}{N_T} = 1 - \frac{N_U}{N_T}, \text{ where:}$$

- $N_S$ – number of successful runs
- $N_U$ – number of unsuccessful runs
- $N_T$ – total number of runs.

Depending on the number of errors remained in programs the number of successful runs can be high or low.

Figure 1 shows how some user actions and inputs can cause activation of existing faults in programs. This will lead to a failure of the programs, and to an unsuccessful run.

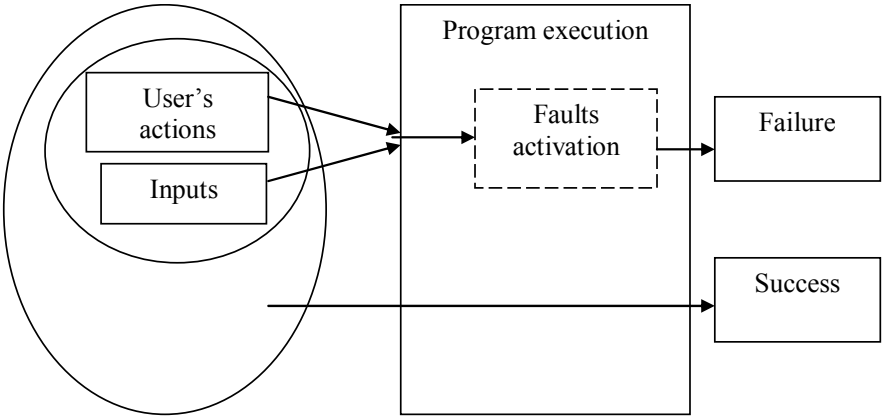


**Fig. 1 Results of programs execution**

The result of a program's execution (success, failure) depends mainly on the number of errors that still exist in the program and on the user actions and inputs that are given.

In an object oriented environment, as many mobile applications are developed, the number of classes written, number of classes instantiated, number of system classes used or inherited are a few of the software metrics that could be used.

## Conclusions

Computing software quality metrics for mobile applications is similar to desktop applications but there are specific issues. Mobile applications' metrics takes into account factors that have to be identified and quantified in order to provide a higher accuracy and to serve as a base for comparison. This is a difficult process, having many types of mobile devices, with different characteristics and development environments. The biggest effort is spent for testing the application. Building reliable software is conditioned by making good testing. As software testing process is good, as customer's costs are lower. This happens also to the producer.

In order to be efficient, the testing process should be followed by good activities of correction and debugging. This will assure that all discovered errors will be corrected and new errors will not be introduced.

## References

[BLOC03] Bloch, Cynthia, Wagner, Annette – *MIDP Style Guide for the Java 2 Platform, Micro Edition*, Addison Wesley, 2003, ISBN 0-321-19801-8

[FOX03] Dan Fox, Jon Box – *Building Solutions with the Microsoft .NET Compact Framework: Architecture and Best Practices for Mobile Development*, Addison Wesley, ISBN: 0-321-19788-7, 2003

[GILL93] Gillies, Alan C. – *Software Quality, Theory and Management*, Chapman & Hall Computing, 1993

[HASH05] Sayed Y. Hashimi, Scott J. Steffan – *Pro Service-Oriented Smart Clients with .NET 2.0*, Apress, ISBN 1-59059-551-3, 2005

[IVAN99] Ivan, Ion, Pocatilu, Paul – *Testarea Software Orientat Obiect*, Editura Inforec, Bucureşti, 1999

[KALI03] Kalinin, Vadim, Rafalovich, Vladimir – *Palm & Pocket PC Programming*, A-LIST Publishing, 2003

[NOŞC03] Noşca, Gheorghe – *Metode şi tehnici pentru optimizarea costului calităţii software*, Teză de doctorat, ASE, Bucureşti, 2003

[PETE00] Peters, James F., Pedrycz, Witold – *Software Engineering – An Engineering Approach*, John Wiley & Sons, Inc, 2000

[POCA03] Paul Pocatilu – *Dictionary Web Service*, in Economy Informatics, vol. III, nr. 1, 2003, pp. 119-122

[POCA04] Pocatilu, Paul, Toma, Cristian– *Dezvoltarea aplicaţiilor mobile in Java*, în Informatica Economica, vol. VIII, nr. 3(31), 2004, pp. 93-97

[POCA05] Pocatilu, Paul – *Mobile Applications Design,* ACTTM International Symposium, Bucureşti, 2005

[POCA06] Pocatilu, Paul – *Mobile Applications' Quality Metrics,* INFOBUSINESS'06, UAIC, Iaşi, 2006

[PRES00] Pressman, Roger S. – *Software Engineering – A Practitioner's Approach, European Adaptation, Fifth Edition,* McGraw-Hill, 2000

[SALM05] Salmre, Ivo– *Writing Mobile Code Essential Software Engineering for Building Mobile Applications*, Addison Wesley Professional, 2005, ISBN 0-321-26931-4

[SOMM01] Sommerville, Ian – *Software Engineering, 6th Edition*, Addison Wesley, 2001

[TEOD99] Teodorescu, Laurenţiu, Ivan, Ion – *Managementul calităţii software*, Editura Inforec, Bucureşti, 1999

[VLIE00] Vliet, Hans van, – *Software Engineering – Principles and Practice,* John Wiley & Sons, 2000