Quality Validation for Mobile Embedded Software

Haeng-Kon Kim[1], Roger Y Lee[2]

[1] Dept. of Computer information & Communication Engineering
Catholic Univ. of Daegu, Korea
hangkon@cu.ac.kr
[2] Dept. of Computer Science, Central Michigan University
U.S.A
lee1ry@cmich.edu

Abstract. Many mobile applications have evolved from simple HTML pages to complex service-oriented applications that have high maintenance cost. This high maintenance cost is due to the heterogeneity of mobile applications, to the fast mobile evolution and to the fast-moving market which imposes short development cycles and frequent modifications. In order to control the maintenance cost, quantitative metrics for predicting mobile applications' maintainability must be used. This study introduces class diagram design metrics based on Conallen's Mobile Application Extension for UML. The study will use data from an industrial mobile application to show the correlation between the class diagram metrics and maintenance effort measured by the number of lines of code changed.
Keywords: mobile applications, metrics, maintainability, UML

1 Introduction

Many mobile applications incorporate important business assets and offer a convenient way for businesses to promote their services through the Internet. Many of these mobile applications evolved from simple HTML pages to complex applications which have high maintenance cost. This is due to the laws of software evolution and to some special characteristics of mobile applications. The following are two software evolution laws [13] that affect the evolution of mobile applications:

• The law of continuing change: A program used in real world must change or eventually it will become less useful in the changing world

• The law of increasing complexity: As a program evolves it becomes more complex and extra resources are needed to preserve and simplify its structure.

In addition to the reasons above, mobile applications have some characteristics that make their maintenance costly such as hypertext structure, dynamic code generation and heterogeneity. In order to control the maintenance of mobile applications, quantitative metrics for predicting mobile applications maintainability must be used.

There are many mobile application metrics that have been proposed in literature [5], [8], [10], [18], [20]. The important point is to validate these metrics to have confidence in them

This paper provides an industrial experiment to show the correlation between class diagram metrics and maintenance effort measured by the number of lines of code changed. The remainder of this paper is organized as follows: Section 2 provides a review on mobile application modeling and introduces the mobile application design metrics. Section 3 describes the industrial case study carried out at a telecommunication company. Finally, section 4 provides a conclusion.

2 Definition of UML class diagram metrics for Mobile Applications

2.1 Mobile Application Modeling using UML

Modeling is a technique used to represent complex systems at different levels of abstraction, and helps in managing complexity. UML is a visual modeling language [4] that can be used to model object-oriented systems. It is possible to use UML to model mobile applications by using extensions supported by UML. Conallen proposed an extension of UML for mobile applications [4].

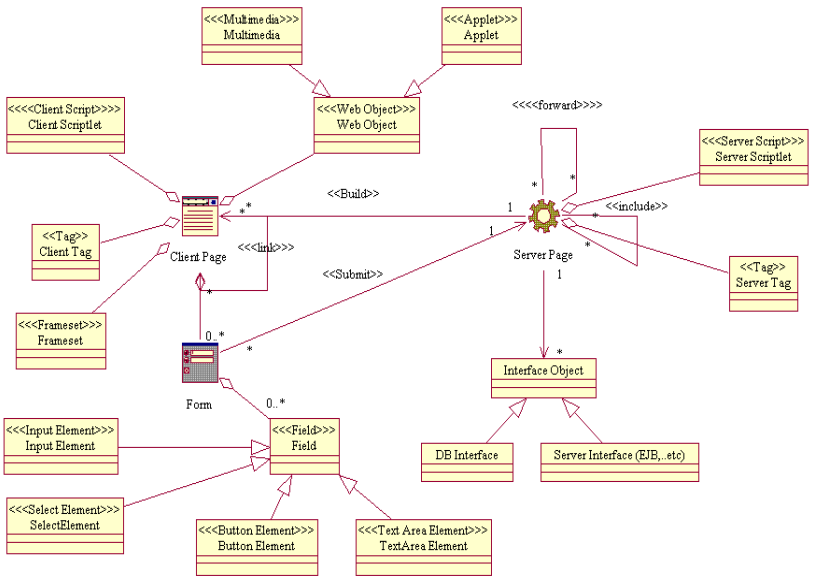

Fig. 1. Mobile Applications Reference Model

Conallen's model defines the following relations as generic associations between different components: builds, redirects, links, submit, includes, and forwards. The builds relationship is uni-directional relationship from the server page to the client page. It shows the HTML output coming from the server page. The redirects relationship is a directional relationship that requests a resource from another resource. The links relationship is an association between client pages and server or client pages. It models the anchor element in HTML. The links relationship can have parameters which are modeled as attributes in the relationship. The submit relationship is a relationship between the form and the server page that processes it. The include relationship is a directional association between a server page and another client or server page. The forward relationship is a directional relationship between a server page and a client or server page. This presents delegating the server request to another page.

The following components are defined as stereotyped classes: Mobile Page, Forms, Components, Scriplets and Framesets. A mobile page is the primary element of a mobile application. It is modeled with two separate stereotyped classes, the client page and the server page. The client page contains client side scripts and user interface formatting. The server page contains server methods and page scoped variables.

Forms are defined to separate the form processing from the client page. The form element contains field elements. Forms are contained in client pages. Each form submits to a different action page. Components run on the client or server page. ActiveX controls and Applets are examples of components. A scriplet contains references to components and controls that are re-used by client pages. A frameset divides the user interface into multiple views each containing one mobile page. Frames can contain more than one client page, but they must contain at least one client page. Conallen's model can describe mobile applications which can be useful during the design phase. Conallen's model has the advantage of being UML compliant and can be extended further to describe mobile applications in a greater detail. In Figure 1 Conallen's model is further extended to include Interface Objects which have an association relationship with the server page.

The Scriplets are divided into server scripts and Client scripts. The server scripts have an aggregation relationship with the server page and the client scripts have an aggregation relationship with the client page.

This study will define a set of metrics based on Conallen's extension of UML. This study will research the relationship between these metrics and maintainability. Maintainability is an important external quality characteristic that can be defined as:

Most studies use source code metrics for measuring maintainability. Some studies in the object-oriented domain measure maintainability using understandability and modifiability [3], [12], [16]. Some of these studies used UML diagrams for measuring maintainability metrics. These models can be used in the mobile domain. However, the UML diagrams must be adapted to the mobile domain by using UML extensions and stereotypes. There are a couple of studies in the mobile domain. Mobile Applica-

tion Maintainability Model (MAMM) [5] is one which used source code metrics and the maintainability was measured using the Maintainability Index. Mendes [17] used design and authoring effort as the dependent variables while the independent variables were based on source code metrics. In [1] design metrics were introduced based on W2000 which is a UML like language. In the study the dependent variables were variations of design effort. The independent variables were measured from the presentation, navigational and information models. For the authors previous research on mobile application maintainability metrics please refer to [7], [9], [11].

3. Case Study

In this section the industrial study is described. This study is an empirical case study conducted at a one of in Korea telecom company.

3.1 Case Study Context

The mobile application used is from the telecommunication Operational Support System (OSS) domain. It is a provisioning application which is used to provision and activate the wireless service in the network. We refer to the mobile application as ProvisionApp. ProvisionApp has around 10,000 users of which 2,500 are concurrent. It is a critical application that is used by customer care advocates to resolve provisioning issues for wireless subscribers. ProvisionApp is built using the latest mobile technologies and frameworks such Struts, and EJBs and uses Oracle for the database. The mobile application uses Java as its main language. It has a Concurrent Versions System (CVS) repository for storing code changes. The application has been operational for 4 years. It has the following five interfaces as shown as shown below:

• SiteMinder: The SiteMinder interface is a plugin to apache server to provide authentication and authorization for the users. The authentication and authorization is done against an LDAP server.

• NMS: The Number Management System (NMS) is the central repository for all wireless phone numbers.

• POM: The Provisioning Order Manager (POM) serves as the work order manager for provisioning data received from the billing applications.

• ESS: The Enterprise Security System (ESS) interface provides the ability for users to initiate Vision Password Reset

• HCM Listener: The Handset Configuration Manager (HCM) application is responsible for over-the-air programming of 3G devices.

This study will use four modules of the ProvisonApp mobile application namely: The Login Module, The Search Module, The Service Trans-action Module, and the Vision Password Module. A brief description of each of the four modules is described in the following sections.

3.1.1 Login Module

Fig. 2. Login Module

The Login Module (Log) shown in Figure 2 provides the ability to access the ProvisionApp by entering a users Network Active Directory domain UserID and Password. Each user can have one of the following three roles:

- Admin: This user profile has unlimited functionality.

- Troubleshooter: This profile has the following capabilities: “View List of Customer Transactions”, “View Customer Transaction details”, “View Network/Device Transactions”, “View Erred Transactions (by interface category) “, “Retry an erred Network/Device Transaction”.

- General User: This profile has the following capabilities: “View list Customer of Transactions “, “View Customer Transaction details”, “View Network/Device Transactions”, “View Network/Device Transaction details “, “Initiate a Vision Password Reset”, “View Vision Password History”, “View Current Network Provisioning Status”.

3.1.2 Search Module

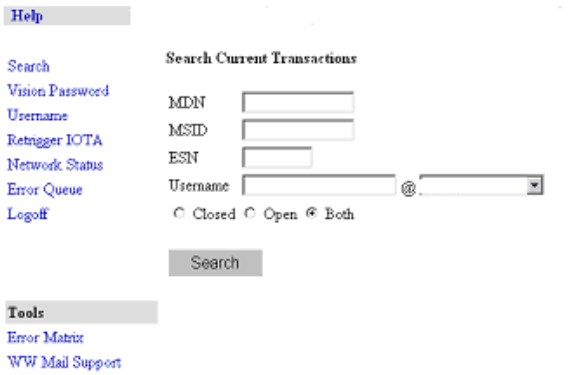

Fig. 3. Search Module

The Search Module (Search) shown in Figure 3 is the first module presented to the User upon successful authentication of the User ID and Password. It provides the ability to Search for transaction by one of the following user identifiers: MDN, MSID, ESN or NAI. It also provides the ability to search Open, Closed or both transactions. Once the parameters are entered and search button is selected the user will be taken to the Current Transactions module.

3.1.3 Service Transaction Module

The Service Transaction Module (Service) provides detailed information about the Service Transaction as shown in Figure 4

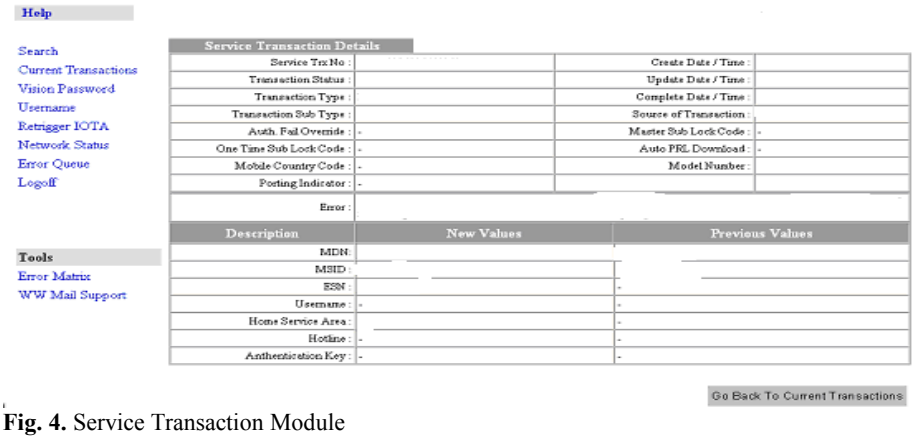

Fig. 4. Service Transaction Module

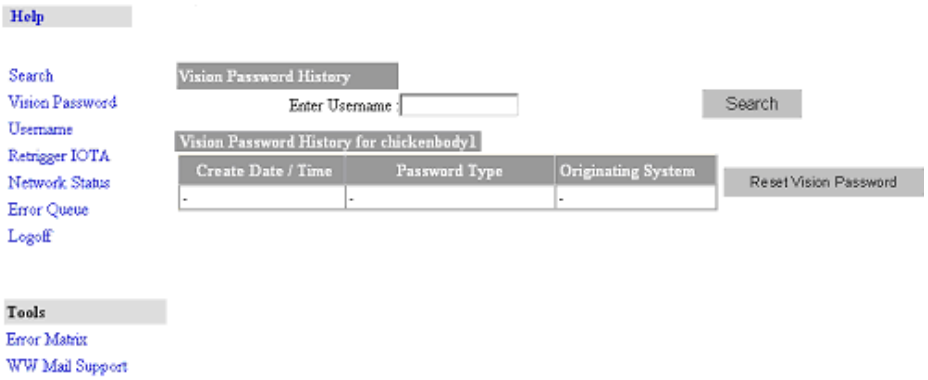

Fig. 5. Vision Password Module

3.1.4 Vision Password Module

The Vision Password Module (Vision) shown in Figure 5 has the following abilities: "Provides Vision Password History for the user with most recent history at top", "Provides the ability to reset the Vision Password for the Username entered", "Vision Password history will not be available until the entire service transaction for Vision Service Activation has completed successfully".

3.2 Hypothesis

This study is trying to test the hypothesis that there is a significant correlation between the current metric set (NServerP, NClientP, NMobileP, NFormP, NFormE, NLinkR, NSubmitR, NbuildsR, NForwardR, NIncludeR, NClientScriptsComp, NServerScriptsComp, MobileControlCoupling, MobileDataCoupling, MobileReusability, NC, NA, NM, NAssoc, NAgg) defined in Table 1 and maintenance effort measured by the number of lines of code changed.

The dependent variable for this study is maintenance effort measured by the number of lines of code changed. The maintenance effort has been used is some studies while validating object-oriented metrics [14], [15].

3.3 Data Collection

In this experiment IBM Rational Rose Enterprise Edition is used to reverse engineer the ProvisionApp mobile application. Rational Rose has a visual modeling component. It can create the design artifacts of a software system. The Mobile Modeler component in Rational Rose supports Conallen's extension for mobile applications. In this

experiment, Rational Rose is used to reverse engineer the four modules of the ProvisionApp mobile application: The Login Module Figure 6, the Search Module Figure 7, the Service Transaction Module, and the Vision Password Module. The modules are all shown in appendix A.

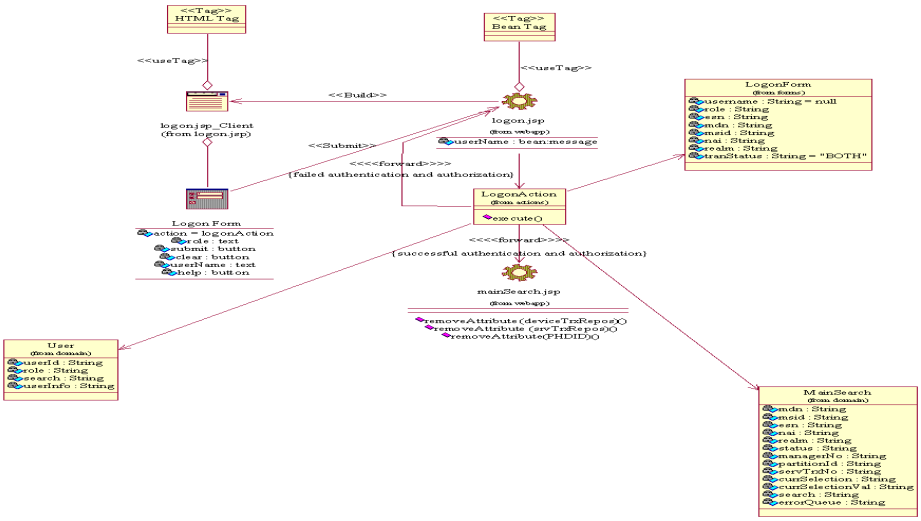

Fig. 6. Login Module Design

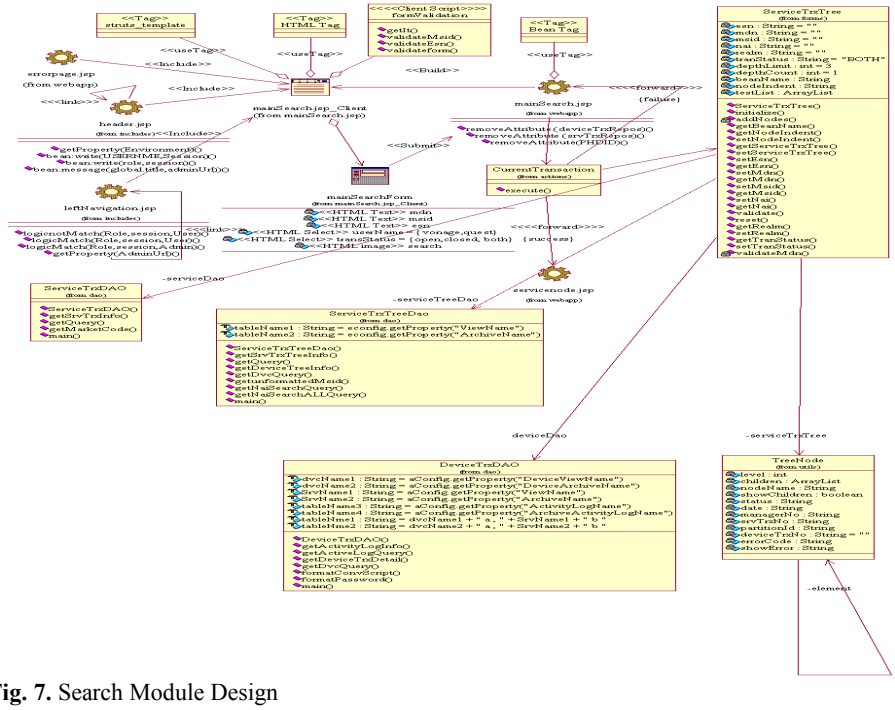

Fig. 7. Search Module Design

3.4 Results and Analysis

The main objective in this study was to explore the correlation between the metrics identified and maintenance effort. The unit of measurement used is Lines of Code changed. The metrics are computed from the class diagrams shown in Appendix A. The maintenance effort is computed from the Concurrent Versions System (CVS) repository. CVS provides a version control system based on open-source code. It keeps track of all work and all changes in a set of files, and allows several developers to collaborate during the software development process.
We have used Statistical Package for the Social Sciences (SPSS) version 14.0 in the data analysis. SPSS is one of the most common computer program used in industry and research institutions for statistical analysis.

Pearson's correlation is used to determine the correlation of the metrics with LOC. Pearson's correlation is a common measure of correlation and reflects the degree of linear relationship between two variables. A correlation of +1 means that there is a perfect positive linear relationship between the metrics and LOC. If the correlation is positive this means that high metric values are associated with high LOC measures. On the other hand, if the correlation is negative this means that high metric values are associated with low LOC measures.

Analyzing the Pearson's correlation shown in Table 1, one can notice that at significance level .05 MobileControlCoupling, NClientScriptsComp, NC, and NM show positive correlation with LOC. The other metrics NServerP, NClientP, NMobileP, NFormP, NFormE, NLinkR, NSubmitR, NbuildsR, NForwardR, NIncludeR, NServerScriptsComp, Mobile-DataCoupling, MobileReusability, NA, NAssoc, NAgg) have a higher significance level than .05. Therefore, their effect is almost 0 to the LOC metric.

Thus, one can say the set of metrics (MobileControlCoupling, NClientScriptsComp, NC, and NM) are correlated with LOC. The other set of metrics (NServerP, NClientP, NMobileP, NFormP, NFormE, NLinkR, NSubmitR, NbuildsR, NForwardR, NIncludeR, NServerScriptsComp, MobileDataCoupling, MobileReusability, NA, NAssoc, NAgg) do not show any correlation with LOC. This result has to be studied in more detail and verified by conducting additional experiments.

As a conclusion, this study can serve as a basis for future studies, and can provide a first indication of the use of the proposed class diagram metrics.

Table 1. Case Study Results

Metric	Significance Level	Correlation with LOC
NServerP	.82	.035
NClientP	.	.
NMobileP	.82	.035
NFormP	.293	.168
NFormE	.293	.168
NLinkR	.175	.216
NSubmitR	.293	.168
NBuildsR	.	.
NForwardR	.	.
NIncludeR	.506	.107
NUseTagR	.949	-.010
NClientScriptsComp	.019	.364
NServerScriptsComp	.418	-.130
MobileControlCoupling	.024	.352
MobileDataCoupling	.828	. -.035
MobileReusability	.175	.216
NC	.019	.366
NA	.361	-.146
NM	.021	.359
NAssoc	.088	.270
NAgg	.175	.216

4. Conclusion and Future Work

This study has introduced metrics for measuring the maintainability of mobile applications from class diagrams. The metrics are based on Mobile Application Extension (MAE) for UML. In this study an industrial experiment at a telecommunication company is carried out in order to show the correlation between the metrics and the maintenance effort. The class diagrams were reverse engineered using Rational Rose and the metrics were measured from the class diagrams. Pearson's correlation was used to analyze the relationship between the metrics and maintenance effort.

The results show preliminary indication of the usefulness of the UML class diagram metrics. The exploratory experiment shows that MobileControlCoupling, NClientScriptsComp, NC, and NM show significant correlation with maintenance effort measured by the number of lines of code changed. In the future we will do more statistical analysis to validate the results further.

5. References

1. Baresi, L., Morasca, S., and Paolini, P.: Estimating the design effort of mobile applications. In: Proceedings of the 9th International Software Metrics Symposium, pp. 62--72. IEEE Computer Society Press (2003)

2. Bhatt, P., Shroff, G., and Misra, A.: Dynamics of software maintenance. In: ACM SIGSOFT Software Engineering Notes, vol. 29, no. 4, pp. 1--5. (2004)
3. Briand, L., Bunse, C., and Daly, J.: A controlled experiment for evaluating quality guidelines on the maintainability of object-oriented designs. In: IEEE Transactions on Software Engineering, vol. 27, no.06, pp. 513--530. (2001)
4. Conallen, J.: Building Mobile Applications with UML. Addison-Wesley (2003)
5. DiLucca, G., Fasolino, A., Tramontana, P., and Visaggio, C.: Towards the definition of a maintainability model for mobileapplications. In: Proceeding of the 8th European Conference on Software Maintenance and Reengineering, pp. 279--287. IEEE Computer Society Press, (2004)
6. Genero, M., Piattini, M., and Calero, C.: Empirical validation of class diagram metrics. In: Proceedings of the 2002 International Symposium on Empirical Software Engineering, pp. 195--203. IEEE Computer Society Press, (2002)
7. Ghosheh, E., and Black, S.: An introduction of UML design metrics for mobile applications. In: Proceedings of the Annual Mphil-PhD Research Workshop, pp. 38--41. Harrow School of Computer Science University of Westminster, (2007)
8. Ghosheh, E., Black, S., and Qaddour, J.: An introduction of new UML design metrics for mobileapplications. International Journal of Computer & Information Science. vol. 8, no. 4, 600--609 (2007)
9. Ghosheh, E., Black, S., and Qaddour, J.: Design metrics for mobileapplication maintainability measurement. In: Proceedings of the 6th IEEE/ACS International Conference on Computer Systems and Applications, pp. 778--784. IEEE Computer Society Press, (2008)
10. Ghosheh, E., Black, S., and Qaddour, J.: An industrial study using UML design metrics for mobile applications. Springer-Verlag (2008)
11. Ghosheh, E., Qaddour, J., Kuofie, M., and Black, S.: A comparative analysis of maintainability approaches for mobile applications. In: Proceedings of the 4th IEEE/ACS International Conference on Computer Systems and Applications, pp. 247. IEEE Computer Society Press, (2006)
12. Kiewkanya, M., Jindasawat, N., and Muenchaisri, P.: A methodology for constructing maintainability model of object-oriented design. In: Proceedings of the 4th International Conference on Quality Software, pp. 206--213. IEEE Computer Society Press, (2004)
13. Lehman, M., Ramil, J., Wernick, P., Perry, D., and Turski, W.: Metrics and laws of software evolution the nineties view. In: Proceedings of the 4th International Software Metrics Symposium, pp. 20--32. IEEE Computer Society Press, (1997)
14. Li, W., and Henry, S.: Object-oriented metrics that predict maintainability. Journal of Software Systems. vol. 23, no. 2, 111--122 (1993)
15. Li, W., Henry, S., Kafura, D., and Schulman, R.: Measuring object-oriented design. Journal of Object-Oriented Programming. vol. 8, no. 4, 48--55 (1995)
16. Mario, M., Manso, E., and Cantone, G.: Building UML class diagram maintainability prediction models based on early metrics. In: Proceedings of the 9th International Software Metrics Symposium, pp. 263--278. IEEE Computer Society Press, (2003)
17. Mendes, E., Mosley, N., and Counsell, S.: Mobile metrics - estimating design and authoring effort. In: IEEE Multimedia, vol. 08, no. 01 pp. 50--57, (2001)
18. Mendes, E., Mosley, N., and Counsell, S.: Early mobile size measures and effort prediction for mobile costimation. In: Proceedings of the 9th International Software Metrics Symposium, pp. 18--39. IEEE Computer Society Press, (2003)
19. Oman, P., Hagemeister, J.: Construction of testing polynomials predicting software maintainability. Journal of Software Systems. vol. 27, no. 3, 251--266 (1994)
20. Ruhe, M., Jeffery, R., and Wieczorek, S.: Using mobile objects for estimating software development effort for mobile applications. In: Proceedings of the 9th International Software Metrics Symposium, pp. 30--37. IEEE Computer Society Press, (2003)