Using Software Quality Standards to Assure the Quality of the Mobile Software Product

Luis Corral

Free University of Bozen/Bolzano
Piazza Domenicani 3, 39100 Bolzano, Italy
Luis.Corral@stud-inf.unibz.it

Abstract

Due to the high relevance gained by mobile software applications, software developers require a way to measure and track the quality of their mobile products from a domain-specific, quantitative point of view. Currently, there is no a link between the general quality goals set by the mobile ecosystem and the practices that have to be exercised to develop a compliant application. In this work, we pursue the implementation of a strategy to extend software quality standards to supply mechanisms to measure the quality of mobile software products from the point of view of developers and users, in the context of mobile execution environments and mobile application markets.

Categories and Subject Descriptors D.2.0 [**Software Engineering**]: General, Standards

Keywords ISO 9126; Markets; Metrics; Mobile; Process; Product; Quality; Standards

1. Introduction

The impact of the mobile software product is growing every day, and has reached a point in which mobile devices have become one of the most important platforms for the distribution and utilization of user-oriented software. Smartphone sales outnumber those of PCs [1], and application markets represent a primary channel for the dissemination of end-user software products, hosting thousands of apps and reporting millions of downloads per day. Handset terminals have experienced a shift from being simple communication devices to become high-end, multipurpose computer equipment. Smartphones are driven by powerful operating systems that allow users to add and remove applications, and they employ architecture that is similar to a regular personal computer. However, these applications have to cope with several constraints inherent to the mobile ecosystem that are not present in conventional desktop computing. These constraints include wireless communication problems, mobility issues, diversity of standards and operating platforms, limited capabilities of terminal devices, security, and many others.

The quality of mobile applications is usually regulated by application market policies, and the perception of quality may also be affected by "word of mouth" via customer's ratings and reviews. Several software lifecycles have been proposed to assist developers in the process of creating successful mobile applications. However, published mobile software lifecycles fail to establish a clear link between the general quality goals imposed upon mobile applications, and specific characteristics that can be measured to determine, the fulfillment of those quality goals quantitatively.

The high relevance gained by mobile software applications triggers the need for a method to measure the quality of mobile software products from a specific point of view.

2. Research Problem

Software applications for mobile devices are created through development practices the objective of which is to create products that are able to perform satisfactorily in a resource-limited environment, while delivering value to the end-users. Additionally, developers must also consider the regulations imposed by distribution channels, which must be met in order to include a product in the application store.

Thus, mobile software engineers should put into practice a development strategy that considers several quality drivers: the mobile environment itself, the expectations of the end-user, and the restrictions set by application markets. A number of software development processes have been proposed to guide software engineers to produce high-quality mobile applications. These processes are typically based on Agile methods, which some consider to be the most effective way to address the diverse and evolving requirements of the mobile environment [2-4].

SPLASH'12, October 19–26, 2012, Tucson, Arizona, USA.
ACM 978-1-4503-1563-0/12/10.

Nevertheless, none of these development lifecycles provide an appraisal model that delivers goals, methods and metrics to directly assess the software product in light of the conditions and constraints given by the mobile execution and marketing environments. Currently, mobile software lifecycles fail to establish an environment-specific measurement plan that links the overall quality goals of the mobile software market, the processes that have to be conducted to develop applications that fulfill such expectations, and the metrics that have to be calculated to quantitatively assure such compliance. We propose to learn from traditional and innovative quality frameworks that have proven their value in other environments and to tailor them to incorporate specific requirements that determine the quality of the mobile product, building upon solid knowledge of the environment, markets and customers.

2.1 Goals of the Research

The objective of this work is to develop the capability to associate the market requirements and success factors of a mobile product with quality characteristics that can be measured. Our work hypothesis states that the mobile software product can be analyzed to relate measurable parameters of the software product with user expectations and market compliance criteria, thus delivering an accurate approximation of the product quality (Figure 1). To achieve these objectives, three research questions were established:

RQ.1: What are the most relevant quality requirements for a mobile application? In other words, what is typically expected from a mobile application from the viewpoint of the involved stakeholder?

RQ.2: What is the impact of the mobile environment on the quality of the software product? This is, what conditions exist in the mobile ecosystem that have a significant impact on the product quality?

RQ.3: How can the quality of a mobile software product be measured and appraised? To determine this, we can utilize the answers from the previous research questions to define a collection of metrics that help to approximate the quality of a mobile product quantitatively.

3. Proposed Approach

Our technical approach is organized in three stages that focus on providing answers for each question established in Section 2.1: First, to identify the most important quality requirements relating to mobile apps, a comprehensive survey of the publishing guidelines of major application stores will be carried out (e.g., iOS store, Google Play, Nook Store, etc.). Publishing guidelines represent the minimum quality expectations for an application to be included in the distribution platform. These conditions make these requirements highly influential factors in the development of mobile applications. We recognize them as instrumental assets in defining what a mobile application should aim to be, what constraints it should observe, and how it should perform. The initial effort concentrates on reviewing and comparing diverse publishing guidelines to extract commonalities, quality factors, and measurable requirements. The next step will be to conduct an analysis that allows us to relate mobile-specific characteristics with their corresponding measurable quality requirements. The mobile-specific target constraints will be studied (e.g., slow processor, limited quantity of memory, limited power resources, etc.). Such study will be carried out through literature review and experimentation (e.g., benchmarking, performance analysis, etc.). This will allow us to isolate additional needs, and track how the mobile environment, by itself, affects the quality of mobile applications. Finally, with the information obtained from the previous stages, the Goal-Question-Metric (GQM) methodology [5] will be used to propose an effective strategy for product quality measurement. So far, a preliminary sketch of our goal has been defined: "To analyze the mobile software product for the purpose of evaluating it with respect to quality; from the view point of developers and customers, in the context of the mobile execution environment and mobile application markets". To describe this goal through questions and to determine the corresponding metrics, the selection of a mature software quality standard is considered, to provide a framework that can be applied to the mobile markets and the mobile ecosystem.Our choice is to revisit ISO/IEC 9126, already suggested for similar domains [6, 7], and to analyze its quality characteristics, focusing on those applicable to the mobile software product, in the given context. ISO/IEC 9126 groups quality characteristics into two orthogonal dimensions: developer's viewpoint (internal and external quality) and customer's viewpoint (quality in use), making it very suitable for the proposed approach. After combining the custom attributes with the core standard characteristics, a robust, mobile-specific quality framework will be available to define quality metrics that can be calculated directly to analyze the mobile software application. In this way, the essential software quality characteristics of ISO/IEC 9126,

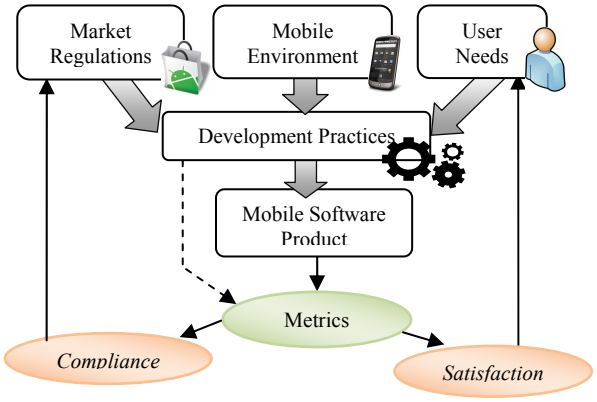

Figure 1. Metric-based Approach to Measure the Quality of Mobile Software Applications

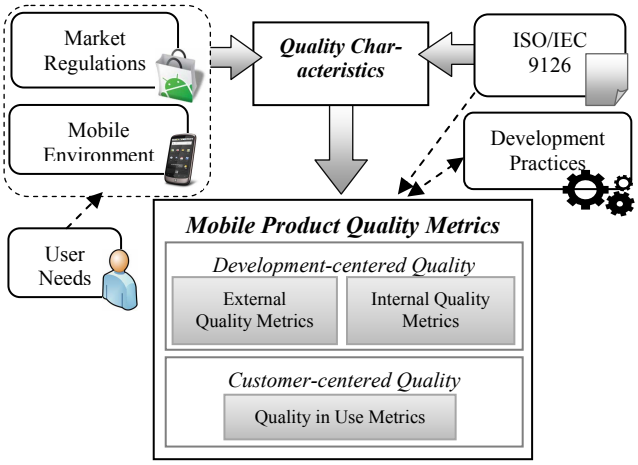

Figure 2. An Approach for Approximating Mobile Software Product Quality Based on ISO/IEC 9126

combined with those identified by us, will be related to metrics capable of delivering quantitative values that accurately approximate the quality of the product in the required contexts, and also to provide information for feedback to the development processes, for continuous improvement (Figure 2).

4. Evaluation Methodology

In accordance with our work hypothesis, we seek to know whether the mobile software product can be analyzed to relate measurable parameters of the software product (i.e., product metrics) with user expectations (e.g., ratings) and application market compliance criteria. To validate our approach, an experiment will be deployed, consisting of the retrieval of an initial sample of 500 assorted applications from Google Play and calculating from them the mobile software quality metrics defined with the GQM methodology. Then, an automatic evaluation will be performed using a non-invasive measurement framework able to analyze the Android-native Java code and calculate user-defined metrics [8, 9]. Using an automatic measurement framework will allow us to execute a multi-dimensional evaluation of the code without adding human-operator noise or bias.

The selected applications should represent a variety of categories (e.g., entertainment, health, lifestyle, etc.) to cover a wider range of user and developer profiles, and they must be distributed under Open Source licenses, so that access to the source code and code repositories can be guaranteed, first to retrieve the code to perform the automatic analysis, and second to study the behavior and evolution of the project (e.g., number of commits, number of releases, etc.) [10]. For each studied application, the product quality metrics obtained after the automatic analysis will be compared to the number of downloads and to the user's rating obtained by the product in the application store, and also with the number of commits and the number of releases stated in the code repository. This evaluation effort will be extended to other mobile operating platforms, in order to analyze the impact of the target platform on the quality of a mobile application, and to understand how the quality metrics of a given mobile app may relate to the constraints imposed by the corresponding application.

5. Conclusions

In this doctoral work, we propose a standard-based strategy for assessing the quality of mobile applications through metrics built upon the demands imposed by application stores and execution environments. Further work will concentrate on isolating the critical quality characteristics from markets and environments to later determine, the metrics that will be used to approximate the quality of the mobile software product. The conduction of the experiment will supply the empirical data required to validate the accuracy of this approach, through the automatic calculation of the metrics and the correlation with product assessment parameters already existing in application stores and code repositories. This study aims to contribute to state-of-the-art providing mechanisms to assess the quality of the mobile software product, based on real-world user expectations and real-world market awareness.

References

[1] International Data Corporation. 2011. Nearly 1 Billion Smart Devices Shipped in 2011. *Release,* retrieved June 7th, 2012. http://www.idc.com/getdoc.jsp?containerId=prUS22917111

[2] Rahimian, V. Ramsin, R. 2008. Designing an Agile methodology for mobile software development: A Hybrid Method engineering approach. *In Proc. 2nd Intl. Conf. on Research Challenges in Information Science.*

[3] Scharff, C., Verma, R. 2010. Scrum to support mobile application development projects in a just-in-time learning context. *In Proceedings of CHASE '10 (ICSE 2010)* pp. 25-31.

[4] Cunha, T.F.V., Dantas, V.L.L., Andrade, R.M.C. 2011. SLeSS: A Scrum and Lean Six Sigma integration approach for the development of software customization for mobile phones, *In Proc. 25th Brazilian Symposium on Software Engineering*, pp. 283-292.

[5] Basili, V. 1993. Applying the goal question metric paradigm in the experience factory. In *Proceedings of the 10th Conference of Software Metrics and Quality Assurance in Industry.*

[6] Spriestersbach, A., Springer, T. 2004. Quality attributes in mobile web application development. *In Proceedings of the 5th PROFES 2004.* LNCS vol. 3009, pp. 120-130.

[7] Mantoro, T. 2009. Metrics evaluation for context-aware computing. *In Proceedings of MoMM2009.* pp. 574-578.

[8] Sillitti, A., Succi, G., Vlasenko, J. 2011. Toward a better understanding of tool usage. *Proceedings of ICSE 2011*

[9] Sillitti, A., Succi, G., Vlasenko, J. 2012. Understanding the impact of pair programming on developers attention: a case study on a large industrial experimentation. *Proceedings of ICSE 2012.*

[10] Sillitti A., Janes A., Succi G., Vernazza T. 2004 Measures for mobile users: an architecture. *Journal of Systems Architecture,* Elsevier, Vol. 50, No. 7, pp. 393-405.