

# Chapter 24

## Requirements Specification as Basis for Mobile Software Quality Assurance

**Raquel Lacuesta**

*Universidad de Zaragoza, Spain*

**Luis Fernández-Sanz**

*Universidad de Alcalá, Spain*

**María del Pilar Romay**

*Universidad San Pablo-CEU, Spain*

### ABSTRACT

*Development of mobile applications is a complex task due to the specific characteristics of the environment. Although different proposals to offer more practical and systematic processes have been published within the so-called mobile software engineering, it is still necessary to devise more advanced methods, especially in the area of software quality assurance. This chapter presents a set of practices to support quality control by developers based on the adoption of methods from traditional software engineering focused on effective requirements specification and associated methods for generating practical testing procedures.*

### 1. INTRODUCTION

Development of software in a mobile context is a really difficult and error-prone task due to the specific characteristics of the environment: hardware limitations, variability of resources, unstable physical and communications conditions, potential security or privacy issues beyond the usual ones in generic networked applications (Choi, 2007),

etc.. Software is frequently the key for success as it is the only component with the necessary adaptability to evolve and decide on conditions where mobile systems should work, although attention and emphasis has been traditionally put into technological issues. This lack of attention has led to ad hoc solutions when developing mobile software so it is clear that a rigorous and systematic approach has to be defined, supported by techniques

DOI: 10.4018/978-1-61520-655-1.ch024

and methods of traditional software engineering adapted to this particular and complex environment. The so-called Mobile Software Engineering is defined as the branch of software engineering which deals with the development and management of mobile software in a real-world mobile scenario, i.e., the management and execution of software within a distributed, evolving context. This means that emphasis is not in the dynamic topology or the details of language design, but on the specific requirements of software within a mobile context.

A detailed review of general software engineering advances reveals that, in spite of its great relevance during the last decade, the “mobile approach” has been seldom used, or it has been tackled in a rather incomplete manner. Certain sub-fields of Software Engineering have paid more attention to this, but not very deeply. Among these we could mention Software Architecture (Capra, Bahsoon, Emmerich, & Fayad, 2008), Formal Languages (Mascolo, Picco, & Roman, 2004; Smith, 2004), Product Lines, Autonomic and Self-Managing Systems (Sakhaee, Leibnitz, Wakamiya, & Murata, 2008), Context-Awareness (Chen & Kotz, 2000), Component-Based and Aspect-Oriented Software Development (Delicato, Santos, Oliveira, Batista, & Pirmez, 2009) and even Model-Driven Engineering (Geihs, Reichle, Khan, Solberg, & Hallsteinsen, 2006). However, the greatest effort has been devoted to techniques and technologies, rather than on methods. Moreover, most of these works have used mobile applications just as an example (Borcea, Gupta, Kalra, Jones, & Iftode, 2007). Much of the research on these lines has focused on these specific topics themselves, and their particular methods, rather than in the mobile aspect of the application being built (Cugola & Jacobsen, 2002). Many of the existing proposals are not clearly emphasizing any Software Engineering techniques, even those which apply them.

One of the most interesting topics (often neglected) is the assessment of the quality of mobile

software applications. Most of the work in literature has been mainly concentrated on quality of service (QoS) (Ayyash, Alzoubi, & Alsou, 2006; Seth, Gupta, & Momaya, 2007; Renesse, Khengar, Friderikos, & Aghvami, 2009). Unfortunately, this has more to do with current levels in communication lines, hardware design and even middleware than with the development of mobile software. In general QoS is being basically assimilated to a philosophy of service-level agreements (SLA), i.e., a minimal performance ratio, particularly on communication issues and on capability to improve response time (Amundsen & Eliassen, 2008; Amundsen, Eliassen, & Agedal, 2009). Therefore, the QoS, understood as an SLA, does guarantee that a set of *non-functional* requirements is fulfilled; but this is not done systematically, using an integral method to assure software quality.

Of course, quality may be addressed by focusing on the development side as well as on the improvement of practical methods for the verification of both functional and non functional requirements. Proposals for making easier and faster the development of mobile applications have been presented in literature during the last years with enhanced environments (Krishna, Sousa, & Satyanarayanan, 2003) or even with software process models, e.g. (Ocampo, Boggio, Münch, & Palladino, 2003; Lutz, 2003). However, what is most covered by contributors are the methods and the solution to technical problems through new artefacts and architectures. Little attention has been paid to the quality side of development in comparison with the rest of the topics; e.g. a typical book like (Salmre, 2005) is not specially focused on quality assurance aspects in a wide perspective as presented in this chapter.

Quality is defined by (International Standards Organisation ISO, 2005) as the features and characteristics of a product or service that bear upon its ability to satisfy stated or implied needs. Obviously, no effective action is possible without a good software requirements analysis. Within mobile technology, especial attention should

12 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/requirements-specification-basis-mobile-software/66480](http://www.igi-global.com/chapter/requirements-specification-basis-mobile-software/66480)

## Related Content

---

### Comprehensive Tool Support for Enterprise Modeling and Evaluation

Patrick Delfmann, Hanns-Alexander Dietrich, Matthias Steinhorstand Jörg Becker (2014). *International Journal of Information System Modeling and Design* (pp. 26-54).

[www.irma-international.org/article/comprehensive-tool-support-for-enterprise-modeling-and-evaluation/119075](http://www.irma-international.org/article/comprehensive-tool-support-for-enterprise-modeling-and-evaluation/119075)

### An Approach Based on Hierarchical Petri Nets for the Verification of Interconnected BPEL Processes

Boukhedouma Saïda and Alimazighi Zaïa (2018). *International Journal of Information System Modeling and Design* (pp. 44-78).

[www.irma-international.org/article/an-approach-based-on-hierarchical-petri-nets-for-the-verification-of-interconnected-bpel-processes/216460](http://www.irma-international.org/article/an-approach-based-on-hierarchical-petri-nets-for-the-verification-of-interconnected-bpel-processes/216460)

### The Impact of Regulatory Compliance on Agile Software Processes with a Focus on the FDA Guidelines for Medical Device Software

Hossein Mehrfard and Abdelwahab Hamou-Lhadj (2013). *Frameworks for Developing Efficient Information Systems: Models, Theory, and Practice* (pp. 298-314).

[www.irma-international.org/chapter/impact-regulatory-compliance-agile-software/76628](http://www.irma-international.org/chapter/impact-regulatory-compliance-agile-software/76628)

### Armadillo Power & Light: A Software Evaluation and Selection Case Study

Louis A. LeBlanc (2001). *Strategies for Managing Computer Software Upgrades* (pp. 44-54).

[www.irma-international.org/chapter/armadillo-power-light/29912](http://www.irma-international.org/chapter/armadillo-power-light/29912)

### Accurate and Language Agnostic Code Clone Detection by Measuring Edit Distance of ANTLR Parse Tree

Sanjay B. Ankali and Latha Parthiban (2022). *International Journal of Software Innovation* (pp. 1-22).

[www.irma-international.org/article/accurate-and-language-agnostic-code-clone-detection-by-measuring-edit-distance-of-antlr-parse-tree/297915](http://www.irma-international.org/article/accurate-and-language-agnostic-code-clone-detection-by-measuring-edit-distance-of-antlr-parse-tree/297915)