

Preparing Mobile Software Development Processes to Meet Mission-Critical Requirements

Position Paper

Luis Corral, Alberto Sillitti, Giancarlo Succi

*Center for Applied Software Engineering, Free University of Bozen-Bolzano
Piazza Domenicani 3, I-39100 Bozen-Bolzano, Italy*

Luis.Corral@stud-inf.unibz.it, {Alberto.Sillitti, Giancarlo.Succi}@unibz.it

1 Introduction

Mobile systems have evolved to a level where they are required to carry out their operations in a wide range of environments, from voice communication, networking and entertainment to complex business applications. Software running on mobile devices has as well grown to a point in which it has earned a major role and impact on the overall performance and capability of a system. This situation attracts a genuine interest of knowing in a more detailed fashion the conditions under which mobile software systems are conceived, designed, implemented, and maintained, both from process and product point of views.

In parallel, “mission-critical” is a recurrent concept in Software Engineering that refers to software systems in which a malfunction may lead to injuries, loss of lives or loss of significant investment. As consequence, these systems require the highest care through all steps of the software development cycle [1]. Application areas of mission-critical software comprehend manufacturing control, transportation systems, weapon delivery systems, space programs, air traffic control, medical systems and many others [2].

The continuous growth in capabilities in software applications running in mobile devices allows them to become an attractive platform to be part of more complex systems, including safety-critical or mission-critical systems. As mentioned above, due to the nature of these environments, such mobile software system would be required to meet strict assurance practices with regard to safety, security and performance, being subject to attain certifications from competent authorities before entering into service.

Currently we count on several software processes and quality management systems intended to support the development of general-purpose (e.g. CMMi, ISO 12207, ISO 15504) or application-specific (e.g. RTCA/DO-178B, ISO 62304, EN 50128) software. These models were conceived and developed without considering restrictions given by a specific target platform, and are general enough to accommodate any execution environments (e.g. embedded, desktop, etc.), so they are prepared to suit the needs and restrictions of mobile equipment.

On the other hand, the number of published development models specific for mobile software systems (e.g. Mobile-D [3] or RaPiD7 [4]) is still limited, and they present addressable shortcomings. These models converge in the selection of Agile methodologies to provide a development framework, calling attention on the necessity of adapting the processes and practices to the evolving needs of mobile software, shorting the development cycles. As a common characteristic, they have claimed little level of criticality on mobile applications, so Agile quality assurance practices do not necessarily fulfill the requirements imposed by application-critical standards or certification procedures [5].

In summary, currently it is not available a methodology or framework to manage the assurance of mobile software so as to produce application-critical, certifiable software, taking into account the needs, boundaries, and particular conditions given by a mobile environment. Such conditions may impact the quality of the software product within a mission-critical system, and deserve to be carefully considered in addition to the practices already dictated by safety-critical software development standards.

We believe that it is necessary to study (i.e. survey, leverage) current software assurance practices adopted by mobile development, and analyze the existing gap between them and mission-critical software standards. After knowing such distance, we may identify the ways in which current practices can be strengthened or adapted to have a robust mobile software assurance process that positively aids mobile developers to produce software able to be incorporated into a certifiable mission-critical environment.

2 State of the Art

It is still difficult to find a disciplined, strong and dependable development process for mobile applications [6]. Abrahamsson et al. [3] discussed that the development of mobile applications is a challenging task due to the specific demands and technical constraints found on a mobile environment. By then, very little was known about the suitability of the different development processes for mobile application development. More recently, in the work of Franke and Weise [7] it is set the question “Does software quality assessment, in particular testing, differ between desktop and mobile software?” after which they evaluate common assurance practices to be applied to mobile software just in the way they are used for other environments.

A first attempt to design a software development methodology for mobile devices is introduced by Abrahamsson et al. [3] in the methodology called Mobile-D (inspired by Extreme Programming). Mobile-D is proposed as an Agile approach to mobile application development, later endorsed by independent works by Abrahamsson [8] and by Rahimian and Ramsin [9]. In the same sense, RaPiD7 (Rapid production of documentation in 7 steps) [4] was developed and applied at Nokia; it is as well inspired by Agile practices and focuses on describing human interaction, planning and documenting in software projects, both related and unrelated to mobile development. In all cases, the incorporation of Agile practices into mobile software processes claimed little criticality level on mobile applications, and a strives for a continuous reduction of the development cycles due to the constant evolution and change of requirements, platforms and users.

These works represent, indeed, an interesting approach to be followed; nonetheless, due to their Agile nature they may fail on establishing a clear software assurance strategy based on traceable plans, procedures, records and reports imposed by application-critical standards and certification authorities, typically based on plan-driven practices.

3 Open Items on Mobile Software Assurance to Meet Mission-Critical Requirements

A significant amount of work is yet to be done to combine mobile software development processes with application-critical standards. The goal is to take advantage of our experience on Agile and mission-critical development to achieve the configuration of a robust framework that learns from both traditional and innovative software engineering practices that have proven their value on the development of conventional and mission-critical software:

- Current mobile software methodologies based on Agile paradigms have to be adapted to meet the needs of application-critical standards (i.e. adding traceability records, documentation, prescribing independence between roles, etc.).
- A number of practices from Agile methodologies (e.g. refactoring, pair programming, customer involvement) should be incorporated to the development of critical software thanks to the added value and positive impact they give to the product’s quality.

The range of applications deployed on cellular telephones and tablets is continuously growing, moving this family of devices towards becoming a primary end-user computing equipment. We are not far from having mobile telephones as software platforms fully operating an aerospace system (e.g. maintenance crews operating handheld gadgets), healthcare systems (e.g. monitors on cellphones with direct communication to hospitals), or similar applications that require to meet safety-critical standards and obtain government certifications previous to enter into service.

4 Vision and Research Approach

Our vision is to innovate, extend and strengthen current mobile software frameworks to assure the quality of the product to the most stringent safety-critical standards and certification requirements. As software for mobile devices grows in capacity and dependability, the lack of a software assurance process for mobile applications is more noticeable, and the unveiling of a proper one becomes an imperative.

To achieve this goal, we have identified as a starting point the adapting of Agile methodologies, based on the fact that such methodologies are now the most accepted approach for managing mobile software development processes. Analyzing independently current mobile software development methodologies and Agile practices, we have identified that they lack of several practices and conditions required by certification processes on application-critical software, for example, rigorous project documentation (plans and results), traceability records, independence between major roles (e.g., independence between development and testing individuals) and other requirements.

Our research approach is to leverage the requirements set by the most important development standards for application-critical software (e.g., RTCA/DO-178B for airborne applications, ISO 62304 for medical systems, and EN 50128 for rail systems) and conduct a gap analysis that will allow us to identify the major limitations shown by current mobile development methodologies with respect to safety-critical standards. In addition, we will incorporate further failure modes given by the mobile environment itself (e.g., failures in communication networks, limited computational resources, and others) to complement the safety considerations to be born in mind when a mobile device and its software are full participants on an application-critical system.

After this analysis, we will count on a series of recommendations (namely practices, records and requirements) to be added to the existing mobile software development methodologies. These recommendations are not intended to replace current development practices for mission-critical software, but they will represent a reliable guideline to strengthen different practices throughout the mobile software development, with the objective that processes, products and documentation are suitable to meet safety-critical certification requirements, and most important, assure the development of a software product that will be deployed in an environment in which there is no room to afford any failure.

References

- [1] McQuaid, P. 2009. *Software Disasters: What have we learned?*. Software Quality Professional, volume 11, issue 3. ISSN 1522-0542. Pp. 28. The American Society for Quality.
- [2] Medikonda, B.S., Seetah Ramaiah, P. 2009. *An approach to modeling software safety in safety-critical systems*. Journal of Computer Science 5, 2009. Pp. 311-332. ISSN 1549-3636. © Science Publications.
- [3] Abrahamsson, P., Hanhineva, A., Hulkko, H., Ihme, T., Jääliñoja, J., Korkala, M., Koskela, J., Kyllönen, P., Salo, O. 2004. *Mobile-D: An Agile Approach for Mobile Application Development*. OOPSLA'04.
- [4] Kylmäkoski, R. 2005. *RaPiD7: A Collaborative Method for the Planning Activities in Software Engineering - Industrial Experiment*. ISBN 952-15-1517-1. Nokia.
- [5] Ambler, S. 2005. *Quality in an Agile World* in Software Quality Professional Journal. volume. 7, issue. 4. ISSN 1522-0542. Pp. 34-40. The American Society for Quality.
- [6] Glissmann, S., Smolnik, S., Schierholz, R., Kolbe, L., Brenner, W. 2005. *Proposition of an M-Business Procedure Model for the Development of Mobile User Interfaces*. IEEE.
- [7] Franke, D. and Weise, C. 2011. *Providing a Software Quality Framework for Testing of Mobile Applications*. Fourth IEEE International Conference on Software Testing, Verification and Validation.
- [8] Abrahamsson, P. 2005. *Mobile software development – the business opportunity of today*. Proceedings of the International Conference on Software Development, 2005. Reykjavik, Iceland, pp. 20-23.
- [9] Rahimian, V. Ramsin, R. 2008. *Designing an Agile Methodology for Mobile Software Development: A Hybrid Method Engineering Approach*. Second International Conference on Research Challenges in Information Science.