Software Development Processes for Mobile Systems

Is Agile Really Taking Over the Business?

Luis Corral, Alberto Sillitti, Giancarlo Succi
Center for Applied Software Engineering
Free University of Bozen/Bolzano
Bolzano, Italy
Luis.Corral@stud-inf.unibz.it, {Alberto.Sillitti, Giancarlo.Succi}@unibz.it

***Abstract*—Mobile applications differ from desktop software due to their particular execution environment, limited resources, high autonomy requirement, market competition, etc. This situation brings the need of having customized development processes that respond efficiently to these challenges, to facilitate the development of high quality products that are able to excel and remain competitive in this domain. While a number of research papers have consistently proposed the adoption of Agile practices, it is not clear how a software development process would help to solve the issues present in the mobile domain. Moreover, there is a lack of evidence that shows a clear link between the proposed methodologies and their utilization in a real-world setting. Finally, the rapid evolution of the mobile environment challenges several of the premises upon which the proposed methodologies were created. In this paper, we present a review on Agile software development processes for mobile applications and their implementations, with the objective of knowing the contribution of Agile methods to address the needs of the mobile software in a production environment. Our goal is to introduce discussion on the need of conducting research that unveils what is the framework of choice of the mobile software industry: if the Agile paradigm was adopted, dismissed, or a new one was created.**

***Index Terms*— Agile, Development, Framework, Mobile, Process.**

I. Introduction

Mobile devices are a key target for software products and services, benefiting from the high impact of ubiquitous computing and the growing capabilities of handheld terminals. The introduction of smartphones triggered the development of a massive number of software products ("apps") that have become extremely popular among users thanks to powerful mobile devices and rich application markets. The amount of software applications for cellular phones and tablets grows at a very fast pace, together with the features and computing power offered by the mobile equipment.

Mobile applications must perform satisfactorily in a heterogeneous and resource limited environment that demands high availability, efficient performance and short response time, while delivering value to the end user. In addition, apps should be developed quickly and should keep a low price to succeed in a market of millions of users exposed to an offer of millions of products. Also, to feature a product in a major application storefront, several market guidelines must be met. These factors build up an environment complex and fault prone that poses upon software engineers the question of how to develop successful products [1].

Software Engineering literature proposes different frameworks for conducting mobile software projects [2, 3, 4, 5, 6], showing a convergent approach based closely on the Agile home ground themes. However, published frameworks do not explain how mobile-specific conditions may impact the selection of a software development process, or why a software development process would help to solve the issues present in the mobile domain. Moreover, there is a lack of evidence about how the proposed methodologies work in a real setting, since the follow up literature tends to be limited and does not provide elements to evaluate their utilization. Finally, the rapid evolution of the mobile environment challenges many of the premises upon which these methodologies were created.

In this paper, we revisit the necessity of having suitable development processes for mobile software, and we present a review on how different Agile-based frameworks claimed to suit the needs of the mobile environment. Our research question is formulated as follows: What is the contribution of Agile methods to address the needs of the mobile software in a real-world environment? To sketch a strategy to solve it, we introduce discussion in four thematic areas: (a) suitability of Agile methods to fit the needs of the mobile business environment; (b) suitability of Agile methods to fit the needs of the mobile operational environment; (c) adoption of the proposed Agile frameworks and evidence about their real use, and (d) the rise of new conditions that challenge some of the premises upon which the proposed methodologies were designed. Our goal is to promote an up-to-date dialogue about the need of conducting research to unveil whether the mobile software industry is really following these paradigms, totally ignored them, or created a different one.

The rest of the paper is organized as follows: Section II revisits the need of having ad-hoc development processes for the mobile product, Section III presents a summary of published Agile methodologies for conducting mobile software projects, Section IV introduces discussion about their real adoption and contribution, as well as the evolution of the mobile domain; to close, Section V outlines further work to keep this research track updated, and draws conclusions

978-1-4673-6333-4/13/$31.00 © 2013 IEEE

II. Mobile Software Development

Like in general-purpose software projects, mobile software development should comply with clear goals and neat practices in order to be successful. Nonetheless, there are several constraints not present in desktop computing that make the mobile ecosystem a particular environment. This includes wireless communication problems (availability, bandwidth variability), mobility issues (autonomy, location-based services), the variety of platforms and technologies, the limited capabilities of terminal devices (low power supplies, small-sized user interfaces), and time-to-market requirements [7].

To organize the universe of constraints that exist on mobile environments, Spataru [8] suggested to classify them in two types: evolving and inherent. Evolving constraints include current limitations that will be solved in the future by the evolution in technology and resources, bandwidth, coverage, etc., (e.g., slow processors or intermittent networks). Inherent constraints are those intrinsic to mobile platforms since they are part of the operational condition of the system and will not be solved in the near future (e.g., a small screen or a limited power source). These constraints should be taken into account by software developers to determine the practices to relieve them as means of non-functional requirements. In addition, mobile software products have a particular business model [9, 10] that foresees high competition, short time to market, and large distribution app stores. Those factors require a link with the development practices that help to produce software able to compete in such an environment [11, 12]. For example, the fast-paced market sets the need of having lightweight processes that facilitate the change and the adoption of new technology or emerging trends. To sum up, an effective development strategy should be strong enough to consider the quality drivers of the mobile ecosystem (such as the environment constraints), the expectations of the end user, and the conditions set by application market. At the same time, it should be aggressive and flexible to adapt to the advancements of the enabling technologies, and to cope with the evolution of the market and the high competitiveness. A solid knowledge on these aspects assists stakeholders to make decisions that deliver better mobile applications, selecting design and development practices that go beyond the utilization of a programming solution [13, 14].

To provide an answer to the discussed challenges it has been consistently proposed that customized Agile practices are the framework of choice for the development of software products for mobile devices. In 2003 it was introduced the discussion on the suitability of Agile methods for the fulfillment of the objectives of the mobile software development [15]. Later, it was shown a thorough mapping between Agile home ground themes with several development traits observed in mobile software [16]. This mapping permitted to outline why Agile methods are a competent solution for implementing development processes in the mobile domain, based on collaboration in small teams, continuous reduction of development times, managing the ongoing changes of requirements, coping with the variety of target platforms, and assuming a small-sized, non-critical product.

III. Agile Software Development Processes for Mobile Applications

Our research question pursues to know the contribution of Agile methods to address the needs of the mobile software in a real environment. To have a better understanding on how Agile practices can be implemented in a mobile project, we surveyed three major digital libraries (IEEE Xplore, ACM DL and ScienceDirect) searching for research articles that present a comprehensive mobile software development framework based upon Agile principles. To make the selection of the mobile development frameworks, we established as criteria: 1) frameworks should be specific and mainly applied for mobile applications; 2) they should focus on the use of Agile practices; 3) frameworks should have been put in practice preferably in at least one study case, and 4) frameworks and case studies must have been published in an international journal or conference proceedings. As result, we found on five Agile approaches: Mobile-D, MASAM, Hybrid Methodology, Scrum, and Scrum Lean Six Sigma (sorted by publication date). We present a summary of each one, and to determine their contribution in a production environment, we looked for instances (research papers, technical reports) that cite these methodologies to illustrate how to utilize them to develop a real-world mobile product. Citation count was taken from the number reported in the digital libraries when available, and complemented from Google Scholar removing duplicates.

A. Mobile-D

Mobile-D was the first attempt to incorporate Agile practices for the development of mobile applications. Mobile-D was introduced in 2004 by Abrahamsson et al. [2] as a development methodology inspired on Extreme Programming, Crystal Methodologies and Rational Unified Process (RUP). It is recommended to be used by a small, co-located team, working in a short development cycle. Mobile-D is structured in five phases (Explore, Initialize, Productionize, Stabilize and System Fix) sequentially arranged. Each phase implies a sprint, covering from stakeholder identification through system test and delivery. Actual activities are founded upon principles aligned with Agile ground practices (e.g., Test Driven Development, Pair Programing, etc.) In spite of its sequential organization, Mobile-D encourages iterations. In general, Mobile-D is the most influential methodology in the field, being cited by 17 articles including case studies, though not all of them referring to mobile development. Other 16 examples of use in real projects are showcased in the Mobile-D website.

B. MASAM

Mobile Application Software development based on Agile Methodology (MASAM) was presented by Jeong, Lee, and Shin [3]. It is based on Extreme Programming, Agile Unified Process, RUP and the Software and Systems Process Engineering Meta-model. The structure and detailed implementation of MASAM show a strong tie with Mobile-D and only introduces minor variations, for example a management tool harnessed on the Eclipse Process Framework. Like Mobile-D, MASAM follows a software life cycle based on the Agile approach. On top of these principles, MASAM proposes a simple life cycle conformed by 4 phases, each one comprising a segment of the development process (Preparation, Embodiment, Development and Commercialization). The implementation of software products is carried out through Test-Driven Development, Pair Programming, Refactoring and Continuous Integration, with iterative testing activities. The insertion of the Commercialization Phase expands the methodology to concentrate on product selling tasks. This methodology does not include a case study, and it is cited by 3 papers, none of which is an experience report of its utilization.

C. Hybrid Methodology Design Process

Rahimian and Ramsin [4] promoted the adoption of Agile and risk-based methodologies as an appropriate framework for the development of mobile systems. It is based on a combination between Agile Methodologies, Adaptive Software Development and New Product Development. To structure their methodology, it is proposed to take as baseline a generic software development lifecycle and customize it with a merge between general Agile practices and principles of methodology engineering for new product introduction. The outcome is the Hybrid Methodology Design Process, organized in iterations that cover from the generation of the idea until the release of the product. It defines an iterative design engine (that designs, models, integrates, and reviews) and market testing and commercialization. Market-awareness principles attempt an efficient commercialization of the resulting product. The research paper that presents Hybrid Methodology Design and its 9 citations do not include a case study or experimental setup that tests this methodology for developing an actual product.

D. Scrum

Scharff and Verma [5] conducted a study to verify the effectiveness of Scrum for the development of mobile applications. Scrum is an iterative and incremental framework commonly used in combination with other Agile practices. It uses iterations of fixed duration (typically one to four weeks) called sprints. At the beginning of each sprint, during the planning, the team commits to complete a certain number of tasks established from the product backlog and documents them in a sprint backlog. After this, the Scrum team decides how much work they will execute in the next sprint. By the end of the sprint, a functional product is delivered, and the pending features (i.e., the new product backlog), will be forwarded to the next iteration. For project tracking, it is used a burn-down chart, that is a relationship between the pending work and the time. The authors propose to use Scrum for mobile software development assuming that phone applications are simple, activity-centered, with a restricted number of actions, and developed by small teams in short periods. This research paper includes one scholar case study and has been cited by 4 articles, including 1 case study not related to mobile software.

E. Scrum Lean Six Sigma

Scrum Lean Six Sigma (SLeSS) [6] is an integration approach of Scrum and Lean Six Sigma for the development of embedded software for mobile phones. This philosophy enables the achievement of performance and quality goals, while progressively improving the development processes in a statistically-controlled basis. SLeSS picks up from the Scrum methodology, pursuing a combination of the effort and consistent deliveries of the Scrum sprints with the continuous process analysis and improvement model represented by the Six Sigma methodology: Define, Measure, Analyze, Improve, and Control. The sprint backlog is used not only to establish the objectives of the next iteration, but it is also carefully examined for statistic-based process improvement purposes. The implementation of SLeSS foresees an incremental approach, in which an Agile approach like Scrum is adapted to coexist with a planned-based methodology like Lean Six Sigma (LSS). Authors propose first to plan the implementation of Scrum alone, after this, it is necessary to apply organizational adjustments to implement LSS. This paper includes a case study of an implementation in a mobile software company, and it is cited by 1 work that is not a report of its utilization.

F. Other relevant works

Not all the mobile-specific software development methodologies are based on Agile methods. We can mention RaPiD7, which is based on Agile principles but focused on the authoring documentation of mobile projects rather than in actual software development [17]. Also, the Mobile Development Process Spiral was proposed to utilize a usability-driven-model to integrate mobile-specific usability matters into existing application development processes. This methodology is not Agile-based, though it supports iterations and practices to ensure that requirements are addressed and validated [18]. Finally, the Intel Mobile Application Development Framework is an enterprise-oriented effort that focuses on evaluating the suitability of mobile applications to generate business value within a company. It prescribes guidance documentation, enabling technologies and supporting resources for conducting projects that adhere better to the organization’s standards and best practices [19].

Another group of works report design practices or implementation guidelines for mobile products. Even though they do not describe a comprehensive software development framework, they provide development guidelines, design methodologies and best practices for mobile software that can be applied following any development process [20, 21, 22, 23, 24, 25]. In most of these works, several of the mobile environment constraints (e.g., memory consumption, energy awareness, user interfaces) are considered to produce guidelines and recommendations for the hands-on implementation of suitable mobile-specific products.

Similarly, a product-oriented family of works concentrates on proposing software metrics specific to mobile applications. These metrics typically consider characteristics particular to mobile applications, and promote the customization of general-purpose product quality standards to suit the needs of mobile environments [26, 27, 28, 29, 30]. Nevertheless, these papers are completely product-focused and do not show the link between the quality goals of the mobile product and the processes that have to be conducted to develop an application that satisfies efficiently these expectations.

IV. Discussion: Accomplishment, Evidence and Evolution

The reviewed methodologies provide different sets of practices that aim to achieve fully functional mobile products on short periods. When available, case studies illustrate the suitability of those methods for assisting the development and delivery of real project. The surveyed methodologies show convergent approaches based on the Agile home ground themes, nonetheless, they leave open how to address the environment-specific circumstances to create more suitable end products. Also, the review shows a lack of supporting evidence about the applicability of the methodologies in a non-scholar setting. Furthermore, there have been significant changes in the conditions of the mobile environment upon which some of the methodologies were developed. For instance, in 2004 concepts like location-based services, app stores, iOS, Android and other current key terms were emerging concepts or did not exist.

While the reviewed Agile-based frameworks deem to suit the needs of the mobile environment, we need further analysis to validate this claim. To outline the strategy to solve this question, we raise discussion in four thematic areas: (a) suitability of Agile methods to fit the needs of the mobile business environment; (b) suitability of Agile methods to fit the needs of the mobile operational environment; (c) adoption of the proposed Agile frameworks and evidence about its real use, and (d) the rise of new conditions that challenge some of the premises upon which the proposed methodologies were designed. To start the discussion, we introduce a preliminary analysis for each one.

A. Are Agile methods the best fit for the needs of the mobile business environment?

Agile methodologies establish their core values in the possibility of focusing on the working product, remaining close to the client, and responding efficiently to the change. Implementing Agile practices allows adapting processes and practices to the unsteady needs of the mobile domain. Mobile apps should be developed quickly and keeping up a low price to be successful in a market of millions of potential users with an offer of millions of products. Agile methods embrace effectively this philosophy to understand the market, structure the product and release it short time frames.

None of the reviewed frameworks question the suitability of Agile methods for the development of mobile apps. Instead, they focus on profiting from their advantages, and when necessary, authors fulfill their shortcomings by including practices from other frameworks. Lately, they even incorporate directions provided by non-Agile development methodologies, using plan-based methodologies or statistical quality control.

B. Are Agile methods addressing the needs of the mobile operative environment?

The reviewed Agile methodologies claim effectiveness on addressing the needs and constraints of mobile software products. However, it is hard to argue for a direct effect on the end product at the level of abstraction on which they are presented in their corresponding papers, mainly process-oriented. Considering the constraints of the mobile environment, it is not clear as to what problems can be injected if general-purpose software development processes are used, or why a software development process would help resolve these issues. For instance, if we consider a scenario in which one has to design an application to work in an environment with poor and intermittent connectivity, what development process (Mobile-D, Scrum, etc.) should be used? Why would the choice be an important factor in the success of the final outcome? Such questions are generally left open at the level covered by software development frameworks.

Instead, an implementation-oriented approach can be found on the range of articles that focus on design practices that concern on solving, from a practitioner approach, the environment conditions that affect the mobile product, including interface design, usability guidelines, human-computer interaction and others. Regarding the mobile environment itself (efficiency, resource utilization, memory and CPU consumption, network utilization and battery usage) development guidelines are proposed at different levels (architectural, design, coding). In this family of works we can also find testing practices, evaluation metrics and other quantitative indicators that can aid the developer to assess the level of accomplishment and suitability of the end product for the execution environment from a practical point of view.

We can suggest that Agile practices suit the business needs of the mobile environment and fall short on providing an implementation framework for the mobile product, but design guidelines move to the driver's seat to furnish mobile-specific development practices that effectively aid to create a product that is suitable for this domain.

C. Is there enough evidence about the real use of the proposed Agile methodologies?

It is not realistic to claim that the frameworks presented are applicable in a real environment without supporting empirical evidence. Besides Mobile-D, the rest of the Agile frameworks cannot argue to have major support on their adoption and implementation in real production environments. Mobile-D keeps records about software projects developed using this methodology, including works carried out both in research and large industry settings (e. g., F-Secure, Nokia, Philips) [31, 32]. Scrum and SLeSS present only one case study as part of the validation of their work, but citing literature does not show further implementations. MASAM and Hybrid do not show a case study, and citing literature does not show any application.

Table I displays a summary of documented instances of implementation of each methodology, including the case studies presented as part of the research paper.

TABLE I. Agile Software Development Processes and Their Implementation

Methodology	Year	Case Studies	Cited by
Mobile-D	2004	16	17
MASAM	2008	0	3
Hybrid	2008	0	9
Scrum	2010	1	4
SLeSS	2011	1	1

We expanded this analysis with a preliminary review of additional field studies[1][2][3] attempting to identify substantiation of the usage of a methodology in real mobile software projects. However, the reviewed studies focus their interest on analyzing the operating system of choice, software development kits, type of applications produced, etc. Although they suggest a clear trend on shortening the development cycle and broaden the impact of the end product, further work is required to unveil the utilization of a consistent development methodology.

D. Have Agile methods kept up with the evolution of the Mobile environment?

When Agile methods were considered the best fit for mobile development, the mobile business and development environment were different to the current one. Table II shows the mapping between the Agile home ground themes and the characteristics of the mobile software, made available in 2005.

TABLE II. MAPPING OF AGILE HOME GROUND THEMES AND MOBILE SOFTWARE DEVELOPMENT [16]

Ideal Agile Characteristic	Mobile Software Development
High environment volatility	Dynamic environment: hundreds of new mobile phones published each year.
Small development teams	Majority of mobile software is developed in micro or SME companies or development teams.
Identifiable customer	Potentially unlimited number of end-users
Object-oriented development	Java and C++ mainly used.
Non-safety critical Software	Majority of existing mobile software is for entertainment purposes. Mobile terminals are not reliable.
Application level software	Mobile applications are stand-alone applications.
Small systems	Size of mobile applications varies but generally they are less than 10,000 lines of code.
Short development cycles	Generally mobile applications and services can be developed within 1-6 month time frame.

Almost a decade of evolution on the mobile domain (software, hardware and business models) has brought significant advancements; therefore the current applicability of this mapping is controversial and invites to conduct an up-to-date discussion. To name some of the differences of the current status of the mobile domain, we identify:

- While hundreds of new mobile models are still released each year, mobile developers also have well settled operating platforms (e.g., iOS or Android), that have solid software development kits (SDK) and APIs that facilitate the interaction with new device models.
- Mobile software is still developed by small teams, but it is also part of the strategy of large corporate teams.
- The range of applications deployed on cellular telephones now includes healthcare monitors or mobile banking apps that are required to meet strict standards to enter into service and cannot be categorized as non-critical software.
- Nowadays, mobile applications spans not only in stand-alone applications but also interacting with other systems, collaboration tools, using heavily network and hardware resources, etc. This also implies that the mobile software product is not anymore small by definition.

Finally, we observe that some Agile-based mobile software development frameworks try to enhance their methods by adapting practices from plan-based methodologies, for example, keeping project documentation, maintaining records, etc. For example, the latest Agile proposal, SLeSS, is an approach to relieve several shortcomings of Agile methods by applying statistical quality control. This represents a complex merge of two different viewpoints: light-weight development practices and a heavy quality control methodology.

Methodologies released more recently have relegated completely the Agile approach [18, 19], reflecting an identified decline on considering Agile practices as a silver bullet [33-38], and instead promoting a savvy strategy to decide when to use which practice evaluating its advantages and disadvantages, to decide how to apply and complement the Agile approach.

V. FUTURE WORK AND CONCLUDING REMARKS

The published development models specific for mobile applications agreed to identify Agile practices as the best fit to conduct a software project in a mobile context. At the level of abstraction presented in the development methodologies, Agile methods seem to bring value helping to keep up with the mobile business trends and marketability, while other mobile-specific constraints like resource limitation are more likely to be managed via development guidelines.

Current scientific documentation does not show a clear link between the proposed methodologies and their utilization in a production setting, and development surveys pay little attention on mobile software development frameworks, leaving open the question whether the Agile paradigm was adopted, dismissed, or the business created a new one. Moreover, the association between the Agile home ground themes and the characteristics of the mobile software upon which these methodologies were built has changed, urging the need of conducting research to have an up-to-date discussion of such relationship.

To provide a final answer to our research question, there is a clear need of conducting evidence-based research that unveils what mobile development practices are actually used, to have a factual picture on how development teams are conducting and managing mobile software projects. These insights may be gained as means of industrial surveys, interviews with mobile software managers and other empirical studies, and from the proactive discussion with the scientific community.

Mobile platforms keep evolving from being a simple communication tool toward becoming the primary end-user computing equipment. Understanding the context of the mobile domain and the best strategies to develop software for this environment is a primary need for researchers and practitioners. Promoting research on mobile software engineering collaborates to furnish better mobile development processes, to produce better applications, and to keep cultivating the knowledge this rich and promising field.

[1] http://www.infoq.com/news/2011/05/A-Survey-on-Mobile-Development

[2] http://www.gqs.ufsc.br/wp-content/uploads/2011/12/GQS-Workingpaper-002-2011-E-v10.pdf.

[3] http://www.comp.nus.edu.sg/~damithch/df/device-fragmentation.htm

REFERENCES

[1] Zhifang, L., Xiaopeng, G., Xiang, L.; Adaptive random testing of mobile application. 2nd International Conference on Computer Engineering and Technology. 2010.

[2] Abrahamsson, P., Hanhineva, A., Hulkko, H., Ihme, T., Jäälinoja, J., Korkala, M., Koskela, J., Kyllönen, P., Salo, O.; Mobile-D: An Agile approach for mobile application development. In proceedings of OOPSLA'04. 2004.

[3] Jeong, Y.J., Lee, J.H., Shin, G.S.; Development process of mobile application SW based on Agile methodology. 10th Intl. Conf. on Advanced Communication Tech., pp. 362-366. 2008.

[4] Rahimian, V. Ramsin, R.; Designing an Agile methodology for mobile software development: A hybrid method engineering approach. 2nd International Conference on Research Challenges in Information Science, pp. 337-342. 2008.

[5] Scharff, C., Verma, R.; Scrum to support mobile application development projects in a just-in-time learning context. In proc. of the ICSE Workshop CHASE 2010, pp. 25-31. 2010.

[6] Cunha, T., Dantas, V., Andrade, R.; SLeSS: A Scrum and Lean Six Sigma integration approach for the development of sofware customization for mobile phones. 25th Brazilian Symposium on Software Engineering, pp. 283-292. 2011.

[7] Rahimian, V., Habibi, J.; Performance evaluation of mobile software systems: Challenges for a software engineer. 5th Intl. Conf. on Electrical Eng., Comp. Science and Control. 2008.

[8] Spataru, A. C.; Agile Development methods for mobile applications. Master Thesis. University of Edinburgh. 2010.

[9] Maurer, F., Succi, G., Holz, H., Kötting, B., Goldmann, S., Dellen, B.; Software process support over the Internet. In proc. of the Intl. Conf. on Software Engineering, pp. 642-645. 1999.

[10] Yamakami, T.; Business model engineering analysis on mobile client-side software platform strategies. 7th International Conference on Mobile Business, pp. 59-64. 2008.

[11] Glissmann, S., Smolnik, S., Schierholz, R., Kolbe, L., Brenner, W.; Proposition of an m-Business procedure model for the development of mobile user interfaces. In proceedings of the International Conference on Mobile Business. 2005.

[12] Hammershoj, A.; Sapuppo, A.; Tadayoni, R.; Challenges for mobile application development. 14th International Conference on Intelligence in Next Generation Networks, pp. 1-8. 2010.

[13] Shiratuddin, N. and Sarif, S.; md-Matrix: Mobile application development tool. In proc. of the International MultiConference of Engineers and Computer Scientists. 2008.

[14] Di Bella, E., Sillitti, A., Succi, G. A multivariate classification of open source developers. Inf. Sciences 221, pp. 72-83. 2013.

[15] Abrahamsson, P., Warsta, J., Siponen, M., Ronkainen, J.; New directions on Agile methods: A comparative analysis. In proc. of the Intl. Conf. on Software Engineering. 2003.

[16] Abrahamsson, P.; Mobile software development: the business opportunity of today. In proc. of the International Conference on Software Development, pp. 20-23. 2005.

[17] Kylmäkoski, R.; RaPiD7: A collaborative method for the planning activities in software engineering: Industrial experiment. ISBN 952-15-1517-1. Nokia. 2005.

[18] Nosseir, A., Flood, D., Harrison, R., Ibrahim, O.; Mobile development process spiral. 7th Intl. Conference on Computer Engineering & Systems, pp. 281-286. 2012.

[19] Doolittle, J., Moohan, A., Simpson, J., Soanes, I.; Building a mobile application development framework. Intel Whitepaper. Available online: http://communities.intel.com/docs/DOC-19555 (Retrieved January 20th, 2013) 2012.

[20] Häkkilä, J., Mäntyjärvi, J.; Developing design guidelines for context-aware mobile applications. 3rd Intl. Conf. on Mobile Technology, Applications and Systems, pp. 1-7. 2006.

[21] Sá, M., Carriço, L.; Lessons from early stages design of mobile applications. 10th Int. Conf. on Human Computer Interaction with Mobile Devices and Services, pp. 127-136. 2008.

[22] Ayob, N.Z., Hussin, R.C., Dahlan H.M.; Three layers design guideline for mobile application. International Conference on Information Management and Engineering, pp. 427-431. 2009.

[23] La, H.J., Lee, H.J., Kim, S.D.; An Efficiency-centric design methodology for mobile application architectures. 7th International Conference on Wireless and Mobile Computing, Networking and Communications, pp. 272-279. 2011.

[24] Park, K.; The 4-tier design pattern for the development of an Android Application. LNCS 7105, pp. 196-203. 2011. Springer.

[25] Valerio, A., Succi, G., Fenaroli. M.; Domain analysis and framework-based software development. ACM SIGAPP Applied Computing Review 5 (2), pp. 4-15. 1997.

[26] Spriestersbach, A., Springer, T.; Quality attributes in mobile web application development. PROFES'04, pp. 120-130, 2004.

[27] Ryan, C., Rossi, P.; Software, performance and resource utilisation metrics for context-aware mobile applications. 11th IEEE International Software Metrics Symposium. 2005.

[28] Dantas, V., Marinho, F., Costa, A., Andrade, R.; Testing requirements for mobile applications. 24th Intl. Symposium on Computer and Inf. Sciences, pp. 555-560. 2009.

[29] Mantoro, T.; Metrics evaluation for context-aware computing. 7th Intl. Conf. on Advances in Mobile Computing. 2009.

[30] Horta-Marinho, E., Ferreira-Resende. R.; Quality factors in development best practices for mobile applications. LNCS 7336, pp. 632-645. Springer. 2012.

[31] Abrahamsson, P.; Agile software development of mobile information systems. In proc. of CAiSE'07. pp. 1-4. 2007.

[32] Hedberg, H., Iisakka, J.; Technical reviews in Agile development: Case Mobile-D. 6th International Conference on Quality Software, pp. 347-353. 2006.

[33] Corral, L., Sillitti, A., Succi, G., Strumpflohner, J., Vlasenko, J.; DroidSense: a mobile tool to analyze software development processes by measuring team proximity. Proceedings of TOOLS 2012, pp. 17-33. LCNS vol. 7304. 2012.

[34] Janes, A.; Succi, G.; The dark side of Agile software development. Proceedings of Onward! 2012. pp. 215-228. 2012.

[35] Moser. R., Sillitti, A., Abrahamsson. P., Succi. G.; Does refactoring improve reusability? 9th International Conference on Reuse of Off-the-Shelf Components, pp. 287-297. 2006.

[36] Sillitti, A., Succi, G., Vlasenko, J.; Understanding the impact of pair programming on developers' attention: a case study on a large industrial experimentation. In proc. of the Intl. Conf. on Software Engineering. pp. 1094-1101. 2012.

[37] Di Bella E., Fronza I., Phaphoom N., Sillitti A., Succi G., Vlasenko J.; Pair Programming and Software Defects – a large, industrial case study. Transaction on Software Engineering, IEEE, to appear (DOI: 10.1109/TSE.2012.68). (n.d.)

[38] Fronza, I., Sillitti, A., Succi, G., Terho, M., Vlasenko, J.; Failure prediction based on log files using Random Indexing and Support Vector Machines. Journal of Systems and Software 86 (1), pp. 2-11. 2013.