Project Management in New Domains through Process-oriented Collection and Analysis of Effort Data

Fabio Bella, Alexis Ocampo, Jürgen Münch

Fraunhofer-Institut für Experimentelles Software Engineering (IESE)

{bella, ocampo, muench}@iese.fraunhofer.de

Zusammenfassung:

Die Planung und die Steuerung von Software-Entwicklungsprojekten in neuen Domänen wie zum Beispiel im Bereich der aufkommenden Wireless-Services stellen eine besondere Herausforderung dar: Zum einen eröffnen neue Technologien Möglichkeiten, deren Grenzen am Anfang eines solchen Projektes unklar sind; zum anderen existiert kein gesichertes Wissen in Bezug auf domänenspezifische Probleme und Risiken sowie auf den für ihre Behandlung benötigten Aufwand. Einige der zu erwartenden Folgen sind unzuverlässige Projektplanung, ungenaue Aufwandsschätzung sowie ungeeignetes Risikomanagement bezüglich der zu verwendenden Prozesse, Ressourcen und Technologien.

In diesem Artikel wird gezeigt, wie der Ansatz der deskriptiven Prozessmodellierung mit einer prozessorientierten Erfassung und Analyse von Aufwandsdaten kombiniert werden kann, um domänenspezifische Probleme und Risiken zu identifizieren. Der kombinierte Ansatz wird anhand von Beispielen aus realen Projekten illustriert. Vorteile sowie potentielle Schwierigkeiten des Ansatzes werden diskutiert.

Schlüsselbegriffe

Erfahrungsbericht, Einführung von Messprogrammen, Projektmanagement, Analyse von Aufwandsdaten, Deskriptive Prozessmodellierung

Abstract:

The planning and control of software development projects in new domains such as the emerging field of wireless services represent a special challenge: On the one hand, new technologies open up possibilities whose constraints are unclear at the beginning of such a project; on the other hand, there is no proven knowledge regarding domain-specific problems and risks as well as the effort needed to deal with those. Some of the consequences to be expected are unreliable project planning, incorrect effort estimation, and unsuitable risk management with respect to the processes, resources, and technologies to be used.

This article describes how the descriptive process modeling approach can be combined with process-oriented collection and analysis of effort data for identifying domain-specific problems and risks. This combined approach is illustrated with examples from real projects. Benefits as well as potential difficulties encountered with the approach are discussed.

1 Introduction

The engineering of wireless Internet services is an emerging application domain characterized by quickly evolving technology, upcoming new devices, new communication protocols, support for new, different types of media, and varying and limited communication bandwidth, together with the need for new business models that will fit in with completely new service portfolios. Examples of new wireless Internet services can be expected in the domains of mobile entertainment, telemedicine, travel services, tracking and monitoring services, or mobile trading services.

Due to its recentness, this domain lacks explicit experience related to technologies, techniques, and suitable software development process models that is based on quantitative data. Unreliable project planning, incorrect effort estimates, and high risk with respect to process, resource, and technology planning, as well as with regard to the quality of the resulting product are inevitable consequences of this lack of experience. One means to capture experience and get valuable insight into a new domain is systematic tracking and observation of representative pilot projects.

This paper presents a study consisting of two case studies that aimed at quantitative baselining. Additionally, the case studies were used to gain qualitative experience. The article aims at giving managers and developers a sense of the behavior of projects in the wireless Internet domain and shows how process modeling techniques can be combined with process-oriented data collection to investigate the peculiarities of new domains.

2 Context of the Study

The article discusses experience gathered through the development of two pilot services within the context of the WISE project (Wireless Internet Software Engineering, No. IST-2000-30028). The project delivered methodologies and technologies to develop services on the wireless Internet. Several partners from different European countries (Finland, Germany, and Italy) were involved in the project.

Pilot service 1 is a solution for real-time stock tracking on mobile devices that allows the user to view real-time quotes. The company responsible for this service is a provider of high-end trading services on the Internet, aimed at banks and brokers. The pilot was the adaptation of an existing Web-based information service. The pilot focused on two challenging requirements: providing a suitable graphical interface to represent the huge amount of data needed by a financial operator to perform an analysis and supporting data transfer based on the push technology. The adoption of the push technology instead of the pull technology was motivated by the fact that frequent refreshing of a large amount of financial data was required and that the end user paid for Internet traffic to and from mobile devices with charging based on data volume and not on connection time.

Three different organizations were involved in the development of the second pilot service, a multi-player online game for mobile devices. It features real-time interaction amongst many players who share a virtual environment consisting of a fantasy world map. Players can collect items and transport them to other locations, chat and fight against enemies and against each other. A player's character can also develop during the game as victories in battle or the completion of special missions brings experience points that can be used to upgrade attributes of speed, strength, dexterity, etc. Games of this kind are popular in the wired Internet world where they are commonly called Massively Multiplayer Online Role Playing Games (MMORPG). Their success lies in the detailed implementation of a huge virtual world shared by many players (in the order of hundreds) who are given just adventure clues without a predefined path to follow.

The pilot services were developed following an iterative and incremental process and both spanned the same development periods. During each of the three iterations, the development focus was placed on a different set of requirements. The first iteration took place during the period December 2001 – December 2002 (13 months), the second iteration during the period February 2003 – October 2003 (9 months), the third iteration during the period November 2003 – September 2004 (11 months).

3 Applied Approach

In the context of the WISE project, a method was developed and used for designing an adaptable software process based on existing practices from similar domains, industrial piloting, and expert knowledge [3]. Figure 1 shows the relationships between the development iterations and the iterative definition and instrumentation of the development process.

At the beginning of each iteration, process-related information is elicited and a new version of the process is modeled and instrumented. During the iteration, quantitative and qualitative data are collected. At the end of the iteration, data are analyzed and interpreted together with the interested parties and applied to produce another version of the process for the next iteration.

As part of the method, experience from related domains was captured through a literature search focused on techniques, tools, methods, case studies, and real projects [11]. At the same time, descriptive process modeling was used in order to capture the real experience of the process performers. Process models for each pilot were built through interviews with the process performers. In order to package experience from the pilots and from similar domains and make it reusable, the descriptive models were integrated into a reference process model [11]. Software processes were modeled with the help of the Spearmint® environment. The architecture of Spearmint® and its features for a flexible definition of views, used for retrieving filtered and specific presentations of process models, is presented in [4]. One distinct view, namely the Electronic Process Guide (EPG), was used for disseminating the reference process information and guiding process performers, since it is Web-based. The EPG facilitated the communication among developers and process engineers in such a distributed development environment (project partners were located in Germany, Italy, and Finland).

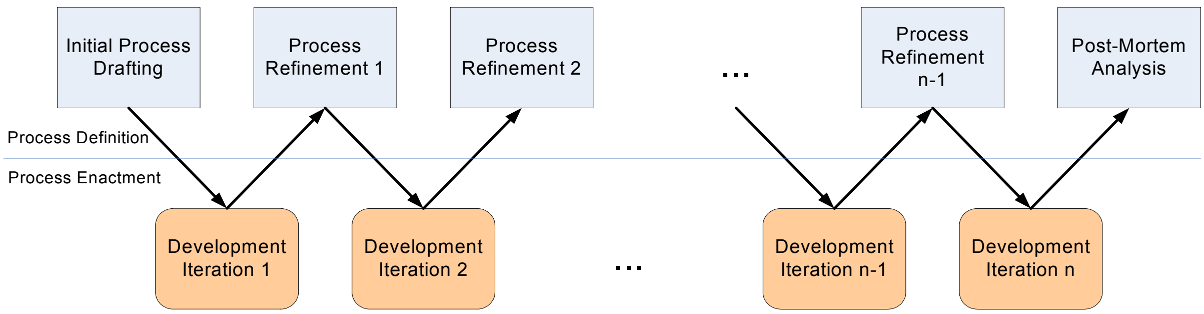

Figure 1: Iterative modeling of the development process

The instrumentation of the process was done with the metrics of the accompanying measurement programs. The goal of the measurement programs was to baseline key figures as effort distribution. The Goal/Question/Metric (GQM) approach was applied to collect process-related quantitative data. Briand et al. [8] describe this approach in terms of six major steps: During the first two steps, business and improvement goals are analyzed and metrics defined according to the process model that describes the whole development process as is; the results of this first phase are GQM plans that comprise all metrics defined. In the following step, the project plan and the process model are used to determine by whom, when, and how data are to be collected according to the metrics: The data collection procedures are the results of this instrumentation. Raw data are collected according to the data collection procedures. Quality models (baselines) are built and analyzed on the basis of the data collected and according to the GQM plans. In the fifth step, the interested parties interpret the baselines and draw possible consequences for the future. Finally, baselines, analysis, interpretations, and consequences are resumed and stored in the experience database for future reuse.

In addition to the measurement of quantitative data, the collection of qualitative data was driven by project retrospectives. Therefore, meetings and interviews with the participants of different work packages were conducted regularly. The key questions proposed by Kerth in [10] were addressed to focus the discussion on learning and improvement: What did we do well, which we might forget if we don't discuss it? What did we learn? What should we do differently next time? What still puzzles us?

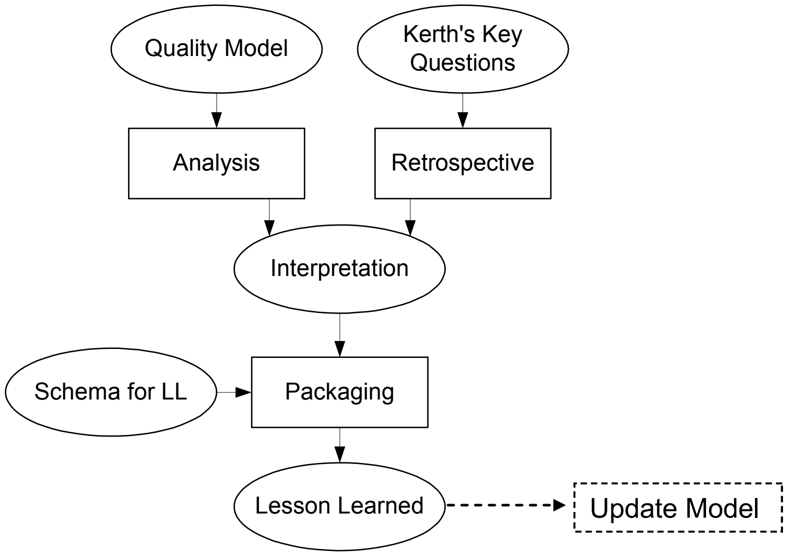

Figure 2: Collection of lessons learned and model update

Figure 2 shows how data collected during each iteration were applied to formalize lessons learned and update the process model.

4 Results

This chapter presents some of the effort baselines gathered during the project together with the main interpretations provided by developers. The chapter is organized around two main sections, one for each pilot service.

4.1 Pilot Project 1

For the analysis, the effort data related to the single activities were consolidated into the three main phases: *Requirements, Development and Test* phase.

Over 750 man-days of effort were spent on the overall development of the service. The greatest effort was spent during the first iteration (about 340 man-days in 13 months), since during this iteration the existing server infrastructure had to be adapted to support the wireless version of the information service. The first version of the wireless service was implemented in WML and derived from an existing HTML version of the service provided over the traditional Internet. During the second iteration, a Java version of the service was created for Blackberry devices. The choice of a full programming language like Java was motivated by the need for push functionality, which was not supported by WML and by the desire to explore Java's graphical features. During the third iteration, the service developed in the second iteration containing Blackberry-specific libraries was modified to be able to run on devices with a color display without the assistance of any proprietary library. The first attempts to run the service on UMTS devices on the Italian market had to be abandoned due to the difficulties encountered when trying to deploy the client application on the devices. In the first half of 2004, only one network provider was distributing UMTS devices in Italy and its policy was to protect them against the deployment of non-trusted applications. Regarding the effort spent on the last two iterations, the amount was comparable if we take into account that the second iteration was shorter than the third iteration (about 200 man-days in 9 months against approx. 220 man-days in 11 months).

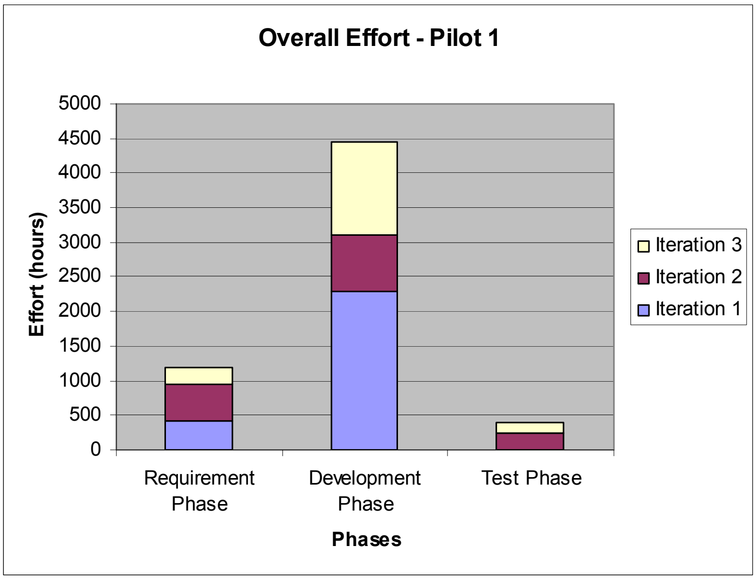

Figure 3: Effort distribution pilot service 1

Figure 3 shows the effort spent on the different phases during the whole project. During each iteration, most of the effort was spent on implementing the service (73% of the overall effort was in the development phase). Relatively little effort was spent during the requirements phase (20%). This was mainly due to two reasons: first, the functional requirements of the service were already known at the beginning of the project, since the same functionality already provided by the existing Web-based version of the service had to be provided on the wireless Internet; second, due to the novelty of the technologies applied and the many non-functional requirements mainly concerned with usability and performance-related issues, requirements could often not be clarified before coding commenced. During the second iteration, greater effort was spent on the requirements phase than in the first iteration (about 65 man-days). One of the main reasons was the introduction of time-consuming feasibility studies performed to explore the capabilities of Java-related technologies. This was deemed necessary when it was observed that during the first iteration a great deal of the effort spent on coding did not produce any tangible result but was absorbed in exploring technology-related issues that could not be understood without resorting to code. Feasibility studies were introduced to explicitly gather this kind of experience and to better understand how the decision-making process concerned with the selection of requirements should be carried out. A total of about 65 man-days - over 8% of the overall development effort - was needed to perform the feasibility studies.

Concerning the test phase, the relatively low amount of effort spent on it - a total amount of about 50 man-days, which was approximately 7% of the overall effort - can be explained by the low complexity of the functions required and the greater priority placed on exploring new technologies than on achieving a version of the service to be considered mature enough for the market. This priority led to the decision to skip the test phase at the end of the first iteration due to deadline pressure. On the other hand, it should also be borne in mind that some effort was spent by third-party organizations on evaluating the service and that this effort was not recorded since it did not happen within the scope of the WISE project. The organizations were interested in buying the client application and evaluated the business model underlying the service as well as the overall usability of the service.

4.2 Pilot Project 2

For the analysis, the effort data related to the single activities were consolidated into four main phases: *Requirements, Design*, *Coding, and Test* phase. To achieve Pilot Service 2, a total of about 680 man-days were required. Most of the effort was spent on coding (55%); design-related activities required about 20% of the work whereas the requirements and the test phases together took up the remaining 25%. In the following, development effort data for the client and the server parts are presented separately.

About 420 man-days were needed to develop the client side of the pilot service. A comparable amount of effort was spent during each of the three iterations. During the first iteration about 140 man-days were spent over a 13-month period, during the second, about 125 man-days over 9 months, and during the third iteration, about 150 man-days over 11 months.

Figure 4 shows the effort data broken down by phases. During each iteration and therefore during the whole project, most of the effort was spent on coding (about 50% of the overall effort), whereas the design phase took the least effort (about 14%). The requirements and the testing phase required a comparable amount of effort (the requirements phase took 80 man-days, 19% of effort, the test phase 70 man-days, 17%).

Looking more closely at these figures, it should be noted that most of the requirements were specified and prioritized during the first iteration. After that, at the beginning of each iteration, the parties involved agreed on a new set of requirements to be addressed and further specification was done verbally and by email. The greater effort spent on the requirements phase during the second and the third iterations can be explained by the introduction of feasibility studies commencing with the second iteration. This was needed since, in a manner similar to Pilot 1, it became clear after the first iteration that many technology-related issues could only be clarified through coding. A total of about 30 man-days (i.e., over 7% of the effort spent on the development of the client side) were needed for feasibility studies.

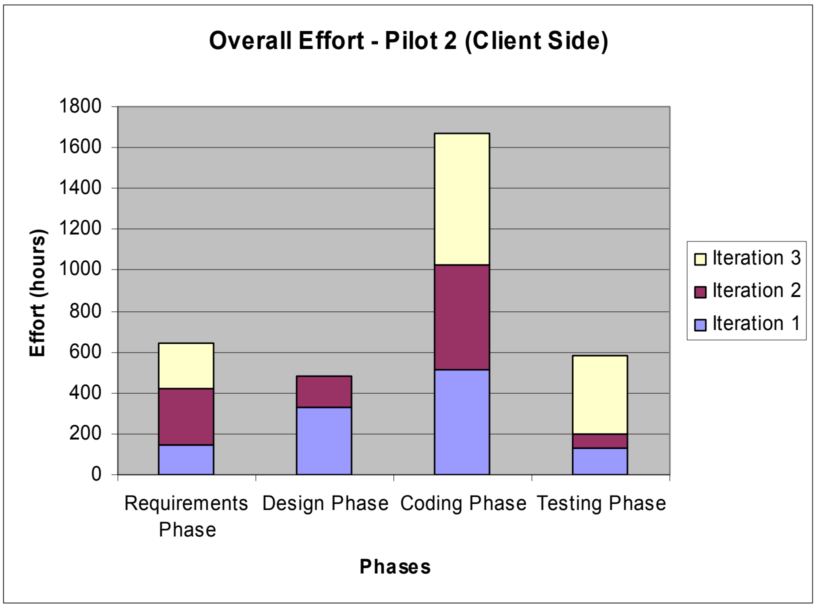

Figure 4: Effort Distribution pilot service 2 - client side

Concerning the design phase, less effort was spent on design in succeeding iterations, explained by the fact that many of the most relevant design-related decisions had already been made. Although design-related activities were performed during the last iteration in order to address design issues concerned with the communication between client and server side, this did not lead to any change in the main design document and the negligible resulting effort was not collected as design-related but as coding-related, since the problems were informally addressed through verbal communication and immediately solved through changes in code.

The coding phase was carried out in a systematic way based on the prioritized requirements and included automated unit testing. A comparable amount of effort was spent on this phase during each iteration of between 60 and 80 man-days per iteration.

About 17% of the effort was spent on the system test phase with the bulk carried out at the end of the third iteration when the whole system was ready. Deadline pressure and the high priority placed on the exploration of available technologies were additional reasons for shifting the main test activities to the end of the project. Another important reason was the intrinsic difficulty of testing applications on mobile devices. The project was torn between the two extremes of either testing on real devices, which was extremely time-consuming due to the lack of automation, or using emulators, which were not reliable substitutes for real devices. As an inevitable consequence, the testing process turned out to be difficult to estimate and to plan. The situation was made even worse if the application had to run and therefore had to be tested on different devices.

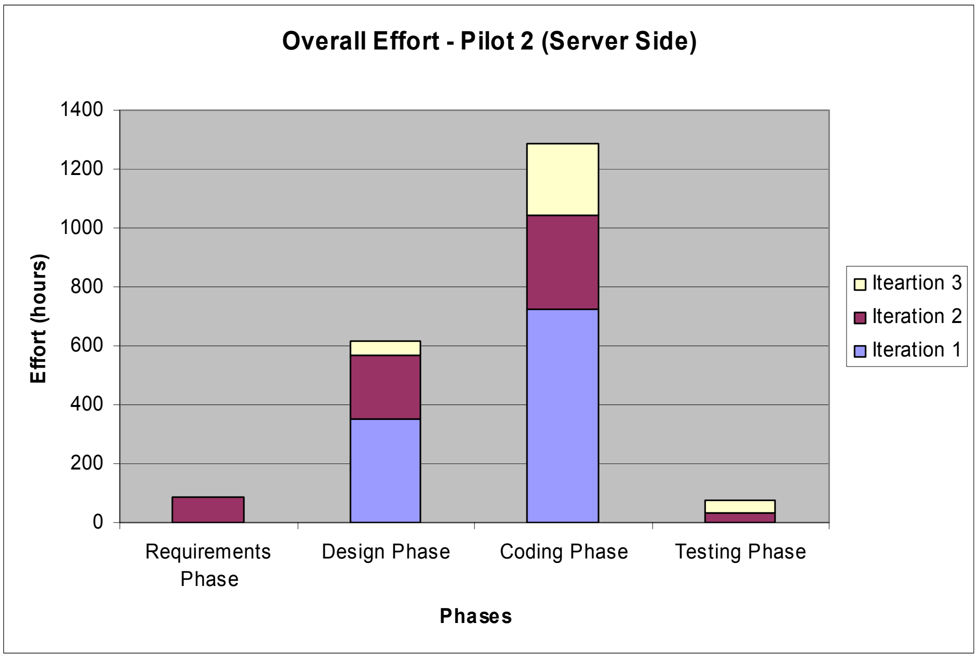

Figure 5: Effort distribution pilot service 2 - server side

Server side development took about 260 man-days. Two different organizations were involved in the development. The organization first responsible for the server left the project after the end of the first iteration and therefore, another partner assumed the role for the remaining two iterations. Figure 5 shows that most of the effort related to the server side was spent during the first iteration (135 man-days, 52% of effort). At the end of the first iteration, the basic infrastructure, an Enterprise Java Beans server, had been set up. The same infrastructure was deployed during the second iteration on servers hosted by the other partner and enhanced during the two iterations to support all requirements formalized at the beginning of the first iteration.

Since the whole development was driven by the organization responsible for the client side, very little effort was spent on the requirements and on the testing phase (about 10 man-days, 4% of effort, per each phase). Similar to the client side development, there was again less effort spent on design in the last two iterations. The main effort was spent on coding-related activities (160 man-days, 62% of effort).

4.3 Discussion

The wireless services discussed represent good examples of the revolutionary applications that will be available on new generation mobile devices. The first service enables stock tracking and exchange in real-time anywhere. The second service allows several players to share the same game environment and introduces the multi-player paradigm to the world of mobile game playing.

In both cases, only a fat-client approach could provide the level of flexibility needed for the development of eye-catching applications characterized by a high level of interaction with the user.

Of course, the data gathered from the development of both pilot services came from specific contexts and cannot be considered universally valid. On the other hand, little data concerning typical effort baselines has been published and, due to the novelty of the field, no data is available that is directly related to the engineering of wireless Internet services. Applying the baselines presented in this article requires a careful comparison of the context surrounding the data discussed and the context surrounding the service to be engineered. Whenever possible, organization-specific historical data should be compared with the baselines presented here and reasons for divergence should be determined. The potential impact of the resulting influencing factors on the new project should be investigated and constantly monitored.

Nevertheless, some aspects seem to be of particular importance since they were observed in both pilot projects and regarded in a similar way by the involved parties. In all cases observed, less effort was spent on design than on coding and most of the effort was spent on coding-related activities. This fact was regarded to be a consequence of the uncertainty inherent in this kind of project and the many unexpected issues that could only be discovered during coding. Feasibility studies took up 8% of the overall effort and this data should be considered an underestimation of the real project behavior, since during the first iteration, the effort spent on feasibility studies was classified as coding. Our experience showed that feasibility studies represent a valid means for investigating requirements and documenting related decisions in the case of the wireless Internet. The percentage of effort spent on testing (between 7% and 12% of the overall effort) would probably turn out to be too low in the case of a mature service to be considered market ready. Testing turned out to be the most difficult phase to estimate and plan due to extremely time-consuming testing on real devices and the unreliability of emulators.

In general, the choice of an iterative and incremental life cycle was considered the only one possible in a field governed by uncertainty. Explicit process modeling together with data collection performed in accordance with the resulting process descriptions represented a viable and fruitful way for gathering high-quality lessons learned, since the facts analyzed during post-mortem sessions were based on measurable evidence. This helped to get a well-founded understanding of project behavior.

5 Conclusions

The proposed combination of descriptive process modeling and process-oriented collection and analysis of effort data could be successfully applied in the context of the WISE project characterized by a great variety of participants in terms of location and process maturity.

The descriptive process modeling approach supported by the Spearmint® environment played a key role in stabilizing the processes, eliciting accurate process models, and disseminating process information to the process performers. These are all necessary preconditions for meaningful effort tracking and planning.

As expected, and in spite of the accurate process models, effort estimation proved to be a challenging process at the beginning. During the first iteration, the organizations involved were not able to deliver effort estimates or the estimates they delivered turned out to be inaccurate at the end of the iteration. On the other hand, effort tracking performed during the first iteration together with estimation processes based on the effort data collected provided more accurate effort estimates for the second iteration. In any case, the collected data represented valuable support for validating and improving the development process. Also, the data helped to identify critical issues and systematically collect lessons learned.

Acknowledgments

We would like to thank the WISE consortium, especially the pilot partners, for their fruitful cooperation. We would also like to thank Sonnhild Namingha from the Fraunhofer Institute for Experimental Software Engineering (IESE) for reviewing the first version of the article.

References

1. Basili, V.R, Quantitative Evaluation of Software Engineering Methodology, in Proceedings of the First Pan-Pacific Computer Conference, Melbourne, Australia (1985).
2. Basili, V.R., Caldiera, G., Rombach H.D.: The Experience Factory, in Encyclopaedia of Software Engineering (John J. Marciniak, Ed.), John Wiley & Sons, Inc., Vol. 1, pp. 469-476 (1994).
3. Becker-Kornstaedt, U., Boggio, D., Muench, J., Ocampo, A., Palladino, G.: Empirically Driven Design of Software Development Processes for Wireless Internet

Services. Proceedings of the Fourth International Conference on Product-Focused Software Processes Improvement (PROFES) (2002).

4. Becker-Kornstaedt, U., Hamann, D., Kempkens, R., Rösch, P., Verlage, M., Webby, R., Zettel, J.: Support for the Process Engineer: The Spearmint Approach to Software Process Definition and Process Guidance. Proceedings of the Eleventh Conference on Advanced Information Systems Engineering (CAISE '99), pp. 119-133. Lecture Notes in Computer Science, Springer-Verlag. Berlin Heidelberg New York (1999).
5. Bella, Fabio; Münch, Jürgen; Ocampo, Alexis: Baselining Wireless Internet Service Development - An Experience Report. In: Brito e Abreu, Fernando (Ed.) et al.: 5th Conference for Quality in Information and Communications Technology, QUATIC 2004 - Proceedings. Porto: Instituto Portugês da Qualidade, 2004, 161-169: Ill., Lit.
6. Bella, Fabio; Münch, Jürgen; Ocampo, Alexis: Capturing Evidence From Wireless Internet Services Development. In: O'Brien, Liam (Ed.) et al.: 11th International Workshop on Software Technology and Engineering Practice. STEP'2003 - Proceedings. Los Alamitos : IEEE Computer Society, 2004, 33-39: Ill., Lit.
7. Bella, Fabio; Münch, Jürgen; Ocampo, Alexis: Observation-based Development of Software Process Baselines: An Experience Report. In: Arbeitskreis Software-Qualität Franken e.V.:CONQUEST 2004. 8th Conference on Quality Engineering in Software Technology - Proceedings. Erlangen, 2004, 31-43: Ill., Lit.
8. Briand, L.C., Differding, C., Rombach, H.D.: "Practical Guidelines for Measurement-Based Process Improvement", Software Process Improvement and Practice 2, No. 4, pages 253-280, (1996)
9. Jedlitschka, A.; Nick, M.: Software Engineering Knowledge Repositories. In: Conradi, Reidar (Ed.) et al.: Empirical Methods and Studies in Software Engineering: Experiences from ESERNET. Berlin: Springer-Verlag, 2003, 55-80: Ill., Lit. (Lecture Notes in Computer Science 2765).
10. Kerth, N.L.: Project Retrospectives: A Handbook for Team Reviews. Dorset House Publishing, ISBN: 0-932633-44-7, New York (2001).
11. Ocampo, A.; Boggio, D.; Münch, J.; Palladino, G.: Toward a Reference Process for Developing Wireless Internet Services. In: IEEE Transactions on Software Engineering 29 (2003), 12, 1122-1134: Ill., Lit.
12. Solingen, Rini van; Berghout, Egon: The Goal/ Question/ Metric Method. A Practical Guide for Quality Improvement of Software Development. London: McGraw-Hill, 1999