Post-deployment Data Collection in Software-Intensive Embedded Products

Helena Holmström Olsson[1] and Jan Bosch[2]

[1] Department of Computer Science, Malmö University, Malmö, Sweden
helena.holmstrom.olsson@mah.se
[2] Department of Computer Science and Engineering, Chalmers University of Technology, Gothenburg, Sweden
jan.bosch@chalmers.se

Abstract. To stay competitive, software development companies need to constantly evolve their software development practices. Companies that succeed in shortening customer feedback loops, minimizing the time between customer proof points and learn from customer usage data will be able to accelerate innovation and improve the accuracy of their development investments. While contemporary research reports on a number of well-established techniques for actively involving customers before and during development, there is less evidence on how to successfully use post-deployment customer data as input to the development process. As a result, companies invest significantly in development efforts without having an accurate way of continuously validating whether the functionality they develop is of direct value to customers once the product is taken into use. In this paper, we explore techniques for involving customers and for collecting customer data in pre-development, during development and in the post-deployment phase of software development. We do so by studying three software development companies involved in large-scale development of embedded software. We present an inventory of the techniques they use for collecting customer feedback and we outline the key opportunities for more effective development and evolution based on post-deployment data collection.

Keywords: Agile development, customer involvement, customer feedback, post-deployment data collection.

1 Introduction

Market uncertainties, competitive pressures, and the constant need for shortened development cycles call for software development practices that are flexible, responsive and adaptive to customers [1, 2]. To respond to this, many software development companies have, since a decade or more, adopted agile practices. In advocating customer involvement and the importance of test-driven development practices [3], agile practices have attracted not only small software development companies, but also companies involved in large-scale development of embedded products.

G. Herzwurm and T. Margaria (Eds.): ICSOB 2013, LNBIP 150, pp. 79–89, 2013.

However, while many companies have succeeded in applying agile practices and, as a result, leveraged the benefits of customer involvement and continuous validation of functionality before and during development, there are few examples of companies that have succeeded in establishing techniques for continuously collecting customer data and validating software functionality also after commercial deployment of the product. The one exception is the Web 2.0 and the software-as-a-service (SaaS) domain where companies like Microsoft [4], Intuit [5] and others, continuously collect customer and product usage data for continuous improvement of the existing product, and as a basis for new product development. Outside of this domain, there is little evidence that software companies have established techniques for collecting and capitalizing on data that is generated after commercial deployment of the product. As a result, companies invest significantly in development efforts without having an accurate way of continuously validating whether the functionality they develop is of direct value to their customers.

In this paper, we present a multiple case study on three companies developing software-intensive embedded systems. While in different domains, all companies aim at continuous collection and analysis of post-deployment data in order to advance their understanding of their customers and how their products are used. The contribution of the paper is twofold. First, it presents an inventory of techniques used for customer involvement and customer feedback collection before, during and after product development. Second, it presents the key opportunities for more effective product development and evolution by collecting customer data in the post-deployment phase of software development.

2 Background

2.1 Agile Software Development

During the last decade agile development methods have dramatically changed the way software development is performed. Agile methods are characterized by short development cycles, close customer collaboration, rapid feedback loops, and continuous evaluation of functionality through test-driven development practices [3, 6]. In comparison to plan-driven development methods, agile methods operate on the principle of "just enough method" and seek to avoid cumbersome and time-consuming processes that add little value to the customer. Although agile methods differ in details and techniques, overall principles such as 'flexibility', 'working code' and 'customer involvement' lie at the heart of all of them. With practices such as daily stand-up meetings, joint planning sessions and project retrospectives, agile methods offer a range of techniques for facilitating collaboration and customer involvement. Lately, test-driven development has become a core agile practice for facilitating continuous validation of functionality. While the agile principles were initially developed for smaller software development organizations, evidence show that large software-intensive organizations operating in complex global development environments are in the process of deploying agile methods as part of their de-facto approach to software development.

However, while agile practices are conducive to close customer collaboration and continuous validation of functionality in the early phases of development [7, 8], there is less evidence on companies that have succeeded in establishing techniques for continuously collecting customer data and validating software functionality also after commercial deployment of the product.

2.2 Customer Involvement

To involve customers in the development process is not a new phenomenon and it is well elaborated upon in user-centered development approaches such as participatory design [10], cooperative design [11], joint-application design [12] and other similar approaches. In these approaches, customers are actively involved in the initial exploration and problem definition phase, as well as during development to help evaluate proposed solutions and different design alternatives. For pre-development involvement, techniques such as use cases, scenarios, prototyping, stakeholder interviews, joint requirements sessions, joint application design sessions etc. are common. Likewise, techniques such as alpha- and beta testing, observation, expert reviews, prototyping, measuring and different types of use cases and scenarios are efficiently used during development in order to continuously validate that the functionality that is developed is of value to the customers. As can be seen in research on agile methods [3, 13, 14, 15], as well as prominent research on requirements engineering [16, 17], techniques such as user surveys, user scenarios, customer archetypes and similar representations are used to capture generic customer needs for mass-market products [17]. Likewise, large-scale agile development often uses product management as a proxy for communicating customer feedback to the development organization before and during development of the product [19].

With regard to post-deployment techniques for customer involvement, the concept of 'lead users' is often used to reflect close collaboration with innovative customers in order to use their feedback for improvement and innovation strategies [20]. As recognized in this research, close collaboration with pro-active customers can result in new functionality as well as new product ideas. Similarly, the 'software ecosystem' approach is referred to as a way for companies to build a community of developers and involve customers to contribute to the improvement activities that take place after product deployment [21]. More recently, the concept of 'Innovation Experiment Systems (IES), and technologies such as Web 2.0, social network systems and Software-as-a-Service (SaaS), or 'on-demand' software, have provided companies with new opportunities to observe and measure system use [4, 5], as well as to run frequent experiments with customers to identify what functionality they value. As reported in relation to these technologies, techniques such as A/B testing (or 'split testing'), customer surveys run by using software tools, analysis of bug reports and product performance data are the most common techniques for continuous collection of post-deployment customer and product data.

In Table 1, we summarize different techniques for involving customers and collecting customer feedback as reported in previous research. As can be seen in the table, there are primarily techniques for this in the early phases of development.

However, techniques for collecting customer feedback after commercial deployment are scarce. While illustrative examples can be found in the Web 2.0 and Software-as-a-Service (SaaS) domain, where sophisticated mechanisms for post-deployment data collection exist, these are not easily applicable for companies involved in large-scale, often embedded, software development.

Table 1. Techniques for customer involvement and customer feedback collection as reported in previous research

Development phase:	**Development activity:**	**Customer involvement /feedback collection technique(s):**
Pre-development	Exploration and problem definition Requirements engineering	Use cases Use scenarios Prototyping Stakeholder interviews Joint requirements development sessions Joint application design sessions Customer representatives/archetypes
During development	Evaluation and validation	Alpha and beta testing Use cases Use scenarios Measuring Prototyping Expert reviews Stand-up meetings
Post-deployment	Evolution and maintenance Improvement and innovation	Lead users A/B testing (split testing) Customer surveys Bug report analysis Performance data analysis

3 Research Site and Method

3.1 Research Site

This paper presents on-going research based on a multiple case study conducted at three software development companies. While the companies differ in domain and size, they are all in the process of establishing mechanisms for collecting customer usage data in the post-deployment phase.

Company A is a provider of telecommunication systems and equipment, communications networks and multimedia solutions for mobile and fixed network

operators. They offer end-to-end solutions for mobile communication and they develop telecommunication infrastructure components for a global market. The company is currently shifting their development practices towards agile development and the notion of cross-functional teams, Scrum practices and continuous testing is well established. For the purpose of this study, we met with key stakeholders at two different company sites:

- *Site 1:* The first site is involved in the development and maintenance of nodes within the 3G networks. At this site we met with a group of four people involving the head of system and architecture, two system managers and a deputy manager.
- *Site 2:* The second site is involved in the development, supply and support of media gateways for mobile networks. At this site, we met with a group of six people involving two department managers, a support manager, a senior specialist, a product manager and an integration leader.

Company B is a manufacturer and supplier of transport solutions for commercial use. The development organization involves coordination of a large number of teams and is largely dependent on supplier organizations. While the majority of the organization is plan-driven in character, there are parts in which agile practices are well established and in which Scrum and XP practices are common practice. For the purpose of this study we met with two attribute leaders, two developers, and one software expert focusing on change management and process improvement.

Company C is world leading in network video and offers products such as network cameras, video encoders, video management software and camera applications for professional IP video surveillance. At the moment, the company is transforming their development practices to reduce lead-time, shorten feedback cycles and increase customer feedback in the development process. For the purpose of this study we met with a group of seven people involving the company CTO, two team leaders, a test manager and two software architects.

3.2 Research Method

Our paper reports on a multiple case study [22] involving three companies involved in large-scale development of embedded software products. The main data collection method used was semi-structured group interviews with open-ended questions [23], with groups of four to seven people. In total, four group interviews were conducted with key people from the different companies. All group interviews were conducted in English and lasted for about two hours. During the interviews, we were two researchers sharing the responsibility of asking questions and facilitating the group discussion to make sure that everyone got a chance to give his/her opinion on the topic. Notes were taken during all interviews and after each interview these notes were shared among the researchers to allow for a discussion regarding the interview session, the answers that had been given and the overall impression of the discussion. As a complement, e-mail correspondence with company representatives was used to clarify any misunderstandings in the transcription of the data.

In terms of data analysis, a qualitative grounded theory approach was adopted [24]. A problem that has been identified in relation to qualitative research is that different individuals may interpret the same data in different ways [25]. This problem was addressed in two ways. First, the grounded theory method prescribes coding processes that provide a traceable, documented justification of the process by which conclusions are reached. Second, we used a 'venting' method, i.e. a process whereby interpretations are discussed with professional colleagues [26]. By sharing notes, and by discussing the results of each group interview, we could develop an accurate understanding of the different contexts.

4 Findings

In this section, we present our interview findings. These are summarized in Table 2 in which we present an inventory of techniques used for customer involvement and collection of customer feedback before, during and after product development. Also, we present the key opportunities for more effective product development and evolution by collecting customer data in the post-deployment phase of software development. The key opportunities were identified during our study and expressed as important by our interviewees when reflecting on ways in which post-deployment data collection can help improve their development practices.

4.1 Pre-development Techniques

Our interviews reveal a range of techniques that are used for involving customers in pre-development activities. In company A, customer contact is channeled through customer units that facilitate communication and coordination between development sites and customers. Twice a year, customer unit workshops are arranged in order for developers and customers to meet. At these workshops a number of different activities are arranged with, e.g., product seminars are held, user groups and test groups meet and lead customers host seminars where specific technology and functionality is discussed. The purpose of these workshops is knowledge sharing and our interviewees report on them as very useful. In addition, customer surveys are used on a regular basis to capture customer experience and satisfaction. At both sites, agile methods are used as the de facto approach to software development and during recent years the product owner role has become the norm for representing the customer in the early phases of development. Together with product management, the product owner works as a 'customer proxy' making sure that customer feedback is reflected in the prioritization of features.

In company B, customer surveys and questionnaires are the most common techniques for involving customers before development starts. Also, the opportunity to meet with customers face-to face is used when possible. In company C, development is based on traditional requirements engineering techniques influenced by customer surveys, product seminars etc. The products are sold to a mass-market and the functionality is prioritized based on generic customer needs.

4.2 During Development Techniques

In all companies, the development phase is characterized by regular customer involvement. In company A (site 1), the development organization is organized so that a number of dedicated teams can work closely with selected customers in order to meet their specific needs. In this way, fast response and customized solutions can be achieved although the customer base is large and dispersed. These customer-specific teams started out as an experiment but have quickly become part of the regular development organization representing an opportunity to increase responsiveness to individual customer needs. In all companies, phone- and videoconferences are common techniques to interact with customers during development. Also, all companies report on site visits as important for increasing the understanding of the customer during development. In company B, the development phase is characterized by constant prototyping, proof of concept and test lab activities. The opportunity to let customers try the vehicles, i.e. test drive, is used when possible. During such test drives important data can be collected and there is the opportunity for customers to communicate directly with the product developers. Safety critical systems as well as functionality related to user experience are constantly tested with customers to validate what functionality they need. In company C, customers are involved on a regular basis by using phone and video conferencing techniques. However, in our interviews at company C we learnt that even though customer representatives are common, developers rarely meet or interact with the customers, or end-users, of their products, and the overall impression was that direct customer feedback was often difficult to achieve.

4.3 Post-deployment Techniques

Based on our interview findings, we see that post-deployment customer involvement in terms of collection of customer usage data is scarce. In company A, both sites report on system operation and performance as types of product data being continuously collected. Also, bug report data is collected in order to learn about system behavior and use. Based on this data, statistical analysis and trend analysis is done and there is the opportunity to learn about current system operation and future dimensioning needs. However, while performance data, such as upgrade success and downtime reports, is collected, company A report on difficulties to use the data. As it seems, customer data is used for trouble-shooting and for maintaining the current version of the product, but very seldom for improving functionality or as a base for developing new functionality. Managers at both sites describe a situation in which data is collected but not used, and they find it difficult to analyze the data to, for instance, learn about what features that are used and what features that are “waste”.

In company B, diagnostic data is collected when the vehicle attends service at an authorized garage. Based on this data, data mining techniques are used to learn about product use. However, while this data is useful for the next iteration of development, i.e. for the next version of the product family, it is collected with very long intervals and is not used for improving the current version of the product or as input to existing functionality. Also, to integrate and to visualize the data is regarded difficult.

In company C, there are no established techniques for post-deployment data collection. While large amounts of data are generated in the systems, these are not used to systematically improve current versions of the systems.

In Table 2, we summarize the techniques that are used for involving customers and collect customer feedback in the different phases of development. As can be seen, the companies have a number of techniques for involving customers in the pre-development and development phase. However, and as recognized in previous research, techniques for post-deployment customer data collection are few.

Table 2. Techniques that the case companies use for customer involvement and customer feedback collection before, during and after product development

Company:	**Pre-development:**	**During Development:**	**Post-deployment:**
A (site 1)	Customer unit workshops Product seminars Surveys Customer representatives	Phone conferences Video conferences Customer-specific teams Stand-up meetings Customer site visits	Product data collection, e.g. system operation/performance data Bug report data collection
A (site 2)	Customer unit workshops User groups Test groups Product seminars Lead customer meetings	Phone conferences Video conferences Customer site visits Stand-up meetings	Product data collection, e.g. system operation/performance data Bug report data collection
B	Questionnaires Interviews Face-to-face customer meetings	Proof of concept Prototyping User test labs Test driving	Diagnostic data collection, e.g. trouble codes, failure reports etc.
C	Surveys Product seminars	Phone conferences Video conferences Customer site visits	Product data collection, e.g. frames per second etc.

4.4 Key Opportunities

While still in the process of establishing techniques for post-deployment data collection, all companies view this activity as critical for continuous validation of their development efforts. In our study, we learnt that there are a number of key opportunities associated with post-deployment data collection. The key opportunities were identified by our interviewees and expressed as important when reflecting on ways in which post-deployment data collection can help improve their development practices. As such, these key opportunities represent the main drivers for more

effective product development and evolution by collecting customer data in the post-deployment phase of software development. The key opportunities are:

- To increase accuracy of development efforts by continuous validation of what functionality customers value.
- To improve requirements prioritization based on customer data.
- To design products that allow for post-deployment data analysis.
- To help customers optimize their use of the product.
- To increase the ability to anticipate future customer needs.
- To increase delivery frequency of functionality.

5 Discussion

The notion of customer involvement, and how to efficiently collect customer feedback, has been a topic of intensive research for decades. However, while there is extensive research on customer involvement techniques for pre-development and development activities [18, 27, 27, 29], research on post-deployment data collection is scarce. Recently, companies such as Microsoft [4], Intuit [5] and others, have adopted techniques to collect customer data after product deployment for continuous improvement of the existing product, as well as for new, innovative product development. While this is a promising area for research, there are few examples that go beyond the Web 2.0 and SaaS domain.

In our study, we explore three companies in the process of establishing techniques for post-deployment data collection. Already, these companies have techniques for exploration and problem definition activities, for capturing customer requirements, and for evaluating and validating proposed solutions and different design alternatives. For example, customer-specific teams are used to increase responsiveness, customer units are used for facilitating customer-developer interaction, and roles such as the product owner are applied to enhance customer impact on feature prioritization [30]. For post-deployment activities such as evolution and maintenance, the companies report on different types of operational data that is being collected. However, even though the companies have data collection mechanisms in place, they find it difficult to integrate, communicate and visualize the data so that it becomes accessible for people in their organization. As a result, post-deployment data is only partially used and the companies agree that there is an untapped potential in customer and product data that is collected after product deployment.

While the companies involved in our study show on post-deployment collection of data, none of them use this data as the basis for improvement of the current product or for innovation of new functionality. This shortcoming is recognized in previous research in which concepts such as ‘online experiments’ [4], ‘test-and-learn’ mind-set [9], and development as ‘innovation experiment systems’ [5] is used to denote techniques that use post-deployment customer data to increase the effectiveness of product development efforts. Inspired by these concepts, our interviewees see a number of opportunities associated with post-deployment data collection. These are related to the ability to increase the accuracy of development efforts, to optimize use of the product, and to increase frequent delivery of customer value.

6 Conclusions

In this paper, we highlight limitations in existent research in terms of post-deployment data collection, and the untapped resource that post-deployment customer and product data remains. Based on a multiple case study at three software development companies, we present an inventory of customer involvement and feedback techniques for pre-development, development and post-deployment activities. Our case study findings confirm previous research in that existing examples of post-deployment data collection techniques are few, and that the data that is collected is mainly used for troubleshooting activities but very seldom for improvement and innovation of products. Furthermore, we identify key opportunities for more effective development and evolution by collecting post-deployment customer data.

Acknowledgements. This study was funded by Malmö University as part of a research collaboration between Malmö University and the Software Center at Chalmers University of Technology and University of Gothenburg, Sweden. We would like to thank the companies involved in the study and the time and engagement allocated by all interviewees.

References

1. Desouza, K., Awazu, Y., Jha, S., Dombrowski, C., Papagari, S., Baloh, P., Kim, J.Y.: Customer-Driven Innovation, Research Technology Management, pp. 35–44 (May-June 2008)
2. Fogelström, N.D., Gorschek, T., Svahnberg, M., Olsson, P.: The Impact of Agile Principles on Market-Driven Software Product Development. Journal of Software Maintenance and Evolution: Research and Practice 22, 53–80 (2010)
3. Highsmith, J., Cockburn, A.: Agile Software Development: The business of innovation, Software Management, pp. 120–122 (September 2001)
4. Kohavi, R., Longbotham, R., Sommerfield, D., Henne, R.M.: Controlled experiments on the web: survey and practice guide. Data Mining and Knowledge Discovery 18(1), 140–181 (2009)
5. Bosch, J.: Building Products as Innovations Experiment Systems. In: Cusumano, M.A., Iyer, B., Venkatraman, N. (eds.) ICSOB 2012. LNBIP, vol. 114, pp. 27–39. Springer, Heidelberg (2012)
6. Hansen, S., Berente, N., Lyytinen, K.: Emerging principles for requirements processes in organizational contexts. Networking and Information Systems 13, 9–35 (2008)
7. Mishra, D., Mishra, A.: Complex software project development: Agile methods adoption. Journal of Software Maintenance and Evolution: Research and Practice 23, 549–564 (2011)
8. Olsson, H.H., Alahyari, H., Bosch, J.: Climbing the "Stairway to Heaven": A multiple-case study exploring barriers in the transition from agile development towards continuous deployment of software. In: Proceedings of the 38th Euromicro Conference on Software Engineering and Advanced Applications, Cesme, Izmir, Turkey, September 5-7 (2012)
9. Davenport, T.H.: How to design smart business experiments. Harvard Business Review (February 2009)

10. Schuler, D., Namioka, A.: Participatory design: Principles and practices. Erlbaum, Hillsdale (1993)
11. Grønbæk, K., Kyng, M., Mogensen, P.: CSCW challenges: cooperative design in engineering projects. Communications of the ACM 36(6), 67–77 (1993)
12. Wood, J., Silver, D.: Joint application development. John Wiley & Sons, New York (1995)
13. Abrahamsson, P., Conboy, K., Wang, X.: 'Lots done, more to do': the current state of agile systems development research. European Journal of Information Systems 18(4), 281–284 (2009)
14. Beck, K.: Embracing Change with Extreme Programming. Computer 32(10), 70–77 (1999)
15. Larman, C.: Agile and Iterative Development: A Manager's Guide. Addison-Wesley (2004)
16. Nuseibeh, B., Easterbrook, S.: Requirements engineering: A roadmap. In: Proceedings of the 22nd International Conference on Software Engineering (ICSE), Limerick, Ireland, June 4-11 (2000)
17. Bennett, K.H., Rajlish, V.T.: Software maintenance and evolution. In: Proceedings of the 22nd International Conference on Software Engineering (ICSE), Limerick, Ireland, June 4-11 (2000)
18. Sommerville, I.: Software engineering, 9th edn. Addison-Wesley, Boston (2010)
19. Larman, C., Vodde, B.: Scaling lean & agile development: Thinking and organizational tools for large-scale scrum. Addison-Wesley (2008)
20. Von Hippel, E.: Democratizing Innovation: The evolving phenomenon of user innovation. Journal für Betriebswirschaft 55, 63–78 (2005)
21. Iansiti, M., Levien, R.: Strategy as ecology. Harvard Business Review 82, 68–78 (2004)
22. Walsham, G.: Interpretive case studies in IS research: Nature and method. European Journal of Information Systems 4, 74–81 (1995)
23. Runesson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. Empirical Software Engineering 14 (2009)
24. Corbin, J., Strauss, A.: Basics of Qualitative Research: Grounded Theory Procedures and Techniques. Sage, California (1990)
25. Kaplan, B., Duchon, D.: Combining qualitative and quantitative methods in IS research: A case study. MIS Quarterly 12(4), 571–587 (1988)
26. Goetz, J., LeCompte, D.: Ethnography and Qualitative Design in Educational Research. Academic Press, Orlando (1984)
27. Beyer, H., Holtzblatt, K.: Contextual design: Defining customer-centered systems. Morgan Kaufmann, San Francisco (1998)
28. Preece, J., Rogers, Y., Sharp, H.: Interaction design: Beyond human computer interaction. John Wiley & Sons, New York (2002)
29. Pressman, R.S.: Software engineering: A practitioner's approach. McGraw-Hill, New York (2010)
30. Schwaber, K., Beedle, M.: Agile software development with Scrum. Prentice-Hall (2002)