Software Development as an Experiment System: A Qualitative Survey on the State of the Practice

Eveliina Lindgren, Jürgen Münch

Department of Computer Science, University of Helsinki
P.O. Box 68, FI-00014 University of Helsinki, Finland
{eveliina.lindgren, juergen.muench}@cs.helsinki.fi

Abstract. An experiment-driven approach to software product and service development is gaining increasing attention as a way to channel limited resources to the efficient creation of customer value. In this approach, software functionalities are developed incrementally and validated in continuous experiments with stakeholders such as customers and users. The experiments provide factual feedback for guiding subsequent development. Although case studies on experimentation in industry exist, the understanding of the state of the practice and the encountered obstacles is incomplete. This paper presents an interview-based qualitative survey exploring the experimentation experiences of ten software development companies. The study found that although the principles of continuous experimentation resonated with industry practitioners, the state of the practice is not yet mature. In particular, experimentation is rarely systematic and continuous. Key challenges relate to changing organizational culture, accelerating development cycle speed, and measuring customer value and product success.

Keywords: Continuous Experimentation · Experiment-Driven Software Development · Customer Feedback · Qualitative Survey

1 Introduction

New possibilities to observe customers and collect customer feedback allow software-centric companies to shorten learning cycles and improve their understanding of customer value. A potential approach is to build software products and services (henceforth, products) by continuously deploying new versions to customers. Instead of relying on pre-defined requirements or opinion-based assumptions, the customer value of products, functionalities, and features is validated in their actual marketplace by conducting a constant series of experiments. This experiment-driven approach is currently most prevalent in the cloud computing environment, but it is beginning to affect the development of all Internet-connected software products [1].

Despite the recent interest in experimentation as an integral part of software development, industrial experimentation experiences have not been studied widely: most examples come from eminent web-facing companies. There has also been relatively little discussion about the obstacles faced by practitioners in this respect.

This paper presents an interview-based qualitative survey that aims at developing an understanding of the state of the practice of using an experiment system approach to software development. Key challenges related to the approach are also identified.

The paper is organized as follows. Section 2 presents the problem background and related work. Section 3 defines the research questions and describes how the study was designed and executed. The results of the study are presented in Section 4, followed by a discussion in Section 5. Finally, the paper is concluded in Section 6.

2 Background and Related Work

During the last decades, agile software development methods have permeated the industry [2]. Agile development has changed the way software is developed for instance by advocating iterative and incremental development, embracing changing requirements, and highlighting the importance of customer feedback. However, Holmström Olsson et al. [3] suggest that the application of agile methods within the research and development (R&D) organization is only one stage on the maturation path of companies' software engineering practices. The following stages are the continuous integration and deployment of R&D output, and finally, R&D as an experiment system. At this stage, development is based on rapid experiments that utilize instant customer feedback and product usage data to identify customer needs.

This final stage is further systematized by Bosch [1]. He emphasizes constantly generating new ideas to test with customers, suggesting that the approach is best described as an *innovation experiment system*. Bosch proposes using 2–4 week R&D iterations followed by exposing the product to customers in order to collect feedback either directly or implicitly by observing product usage. Various experiment techniques can be used throughout development. Experimentation does not necessarily require functioning software. Furthermore, the scope of experiment-driven development can vary from new products to new features and feature optimization.

Fagerholm et al. [4] combine the above-mentioned ideas with key elements from the lean startup methodology [5] and propose a framework for *continuous experimentation*. Continuous experimentation refers to the constant testing of the value of products as an integral part of the development process in order to evolve the products towards high-value creation. Consecutive iterations of the Build-Measure-Learn feedback loop structure the development process. Within each Build-Measure-Learn block, "assumptions for product and business development are derived from the business strategy, systematically tested, and the results used to inform further development of the strategy and product" [4]. This experiment-driven learning process is supported by a technical infrastructure that 1) enables the lightweight releasing of minimum viable products (MVP), 2) provides means for advanced product instrumentation, and 3) supports the design, execution, and analysis of experiments. Fagerholm et al. also provide a description of the roles, tasks, and information artefacts that are required to run a system of continuous experimentation.

Several case studies on companies' experimentation experiences have recently been published. Microsoft's experiences with systematic large-scale online controlled experiments have been recounted in numerous reports, for instance [6]. Google purports to experimentally evaluate almost every change that has the potential to affect user experience [7]. Supporting and fostering continuous innovation is a key element of the Google experiment system [8]. The Netflix “consumer data science” approach is two-staged [9]: experiments are first conducted offline, and if they succeed, an online customer experiment is executed to provide definitive validation.

Adobe’s “Pipeline” innovation process attempts to maximize the learning about a given problem through rapid prototyping and frequent customer evaluation [10]. eBay uses a multitude of experimental techniques in addition to online controlled experiments, such as usability testing, focus groups, and diary studies [11]. The diverse experimentation practices of Intuit are described in [1].

The use of product usage data in the embedded systems domain is examined in [12,13]. The papers conclude that product usage data is not utilized efficiently as a basis for product improvements and innovations. Finally, examples of successful experimentation experiences in an academia-industry collaboration setting are described in [4], [14].

The above-mentioned studies portray different approaches to experimentation. In the context of this paper, the following criteria are used as requirements for *systematic experimentation*: 1) the business-driven definition of explicit assumptions, 2) the design and conducting of experiments to test those assumptions, 3) the analysis of experiment data, and 4) the use of experiment results as input for decision making and follow-up action. Continuous experimentation is achieved if these steps are a permanent part of the development process.

3 Study Approach

Research Questions. Based on the study goals, the following research questions were defined:

RQ1: How is continuous experimentation applied in software development companies?

RQ1.1: How is customer feedback concerning the software product collected?

RQ1.2: How is the collected customer feedback used in the software product development process?

RQ2: What challenges are associated with continuous experimentation?

Study Design. The study was founded on a qualitative survey design, using interviews with industry practitioners to collect data [15,16]. Methodologically, qualitative surveys resemble multiple case studies [16,17]. However, while multiple case studies aim to produce an in-depth analysis of particular cases, the focus of qualitative surveys is less specific and more concerned with providing a multifaceted, diverse view of the topic of interest [15,16].

Semi-structured individual interviews were used to collect data, since they enable focusing on predefined research topics while also being highly flexible to allow for unforeseen information [18]. To structure the interviews, an interview guide was developed, outlining the key topics, questions, and prompts. Easy “warm up” and “cool down” questions were asked at the beginning and end of the interviews. The main topics of the interviews, along with example questions, are defined below (the complete interview guide is available in the Figshare repository [19]):

1. Current software development practices
    a. What kind of software development process do you use?
2. Current practices of customer feedback elicitation and use
    a. How do you make sure that you are building the right product?
    b. How do you collect customer feedback?
    c. Do you collect data about customer behavior, for example in the form of product usage data?
    d. How do you use the collected customer feedback and other data?
3. Future practices of customer feedback elicitation and use
    a. Do you think your current practices of customer feedback collection and customer involvement are adequate?
    b. Are there any obstacles to obtaining deeper customer insights?

The interview data was examined through thematic coding analysis [18]. The analysis was based on an iterative coding process, during which a hierarchical codebook was developed inductively based on the interview data. Descriptive, analytic, or category marker codes were generated depending on the analytic needs. The codebook is also available in the Figshare repository [19]. The codes were then combined to identify common themes, or patterns, within the data.

A purposive, non-probability sample [15,16] was chosen for the study. Software development companies of various sizes, domains of operation, and stages of life cycle were sought to achieve a diverse set of participants. Furthermore, interviewees from different roles and with solid experience in the software industry were sought.

Study Execution. Study participants were recruited among the affiliates of the Need for Speed research program [20] and, outside the research program, through the professional contacts of the authors. Due to practical constraints, only companies operating in Finland were considered. Gatekeepers were contacted at each company, who either participated in the study themselves or suggested a suitable interviewee. In accordance with ethical guidelines [21], the purpose and procedures of the study were shared with the participants via an information sheet, in addition to which they were asked to give voluntary informed consent to partake in the study.

The recruitment resulted in the participation of ten software companies, represented by thirteen interviewees. The individual interviews were conducted in Finland between February and April 2014. The average length of the interviews was 48 minutes, with the range spanning between 36 and 64 minutes. All interviews were conducted in English and audio-recorded to allow for accurate, detailed data analysis. Eleven interviews were conducted by one researcher, and in the remaining two cases two researchers were present. Eleven interviews were performed face to face on the interviewees’ company premises, one via video conferencing, and one as a VoIP call.

To facilitate data analysis, interview recordings were transcribed verbatim shortly after each interview. The transcripts were coded and analyzed using ATLAS.ti [22].

4 Results

This section first gives an overview of the study participants. It then outlines the software development practices of the participating companies. The companies' practices of eliciting and using customer feedback are considered next, after which the challenges with relation to continuous experimentation are presented.

4.1 Overview of Participants

Ten ICT companies operating in Finland participated in the study. The focus was on their software product development functions. Table 1 gives a characterization of the companies by size, domain, and product orientation (more details are not disclosed due to confidentiality reasons). Three of the companies can be described as startups.

Most interviewees held either senior management (31%) or middle management (54%) positions in their companies. Consultant and senior software architect roles were also represented (15%). The interviewees' length of employment in their current company varied between 1 and 26 years, with the average being 7.7 years.

Unlike the other companies who only had one representative, company C was represented by four interviewees. Their answers were merged together to form an overall impression of the company. As regards company E, their software development practices were not discussed during the interview since the interviewee was not actively involved in this part of the company's operations. Input from company E is therefore only included in the results presented in Section 4.4.

Table 1. Participating companies (size classification: small < 50, medium ≤ 250, large > 250)

Company	Company size by no. of employees	Company domain	Product orientation
A	Small	Gaming	B2C
B	Small	ICT services	B2B
C	Large	ICT services	B2B
D	Small	Sports	B2B, B2C
E	Medium	ICT services	B2B
F	Small	Software development tools	B2B
G	Medium	Software development tools	B2B, B2C
H	Large	Security	B2B, B2C
I	Large	Telecom	B2B
J	Small	Multimedia	B2B

4.2 Software Development Practices

All companies mentioned that they utilize agile methods such as Scrum, Kanban, and Lean. All companies also stated that they use continuous integration (CI) but, consistent with previous research [23], there was variability in how CI was interpreted and implemented. These findings are based on the interviewees' informal descriptions of their development approach rather than a formal questionnaire or definition provided by the researchers. The general impression of the companies' development practices was similar to a recent survey [2], although the prevalence of lean-inspired practices and CI appeared to be higher.

Release cycle length ranged from under one month (56%) to less than three months (33%) or more (11%). Interviewees often made remarks on constantly having a deployable product version available, working in a production-like environment to simplify deployments, and pursuing a DevOps mode of operation. The overall impression was that deployments were quite lightweight and flexible, except for on-premises installations in business-to-business (B2B) environments.

4.3 Practices of Eliciting and Using Customer Feedback

The companies used a wide array of techniques to learn about customer needs. Most techniques were based on eliciting direct customer feedback through familiar means such as stakeholder interviews and surveys, prototypes, usability and user experience testing, and other forms of user testing. Bug reports and feature voting were also used as a way to guide development.

Implicit customer feedback in the form of product usage data was collected by five companies (55%). In many cases the product instrumentation covered performance data and basic user demographics. However, some companies also had more sophisticated, feature-level instrumentation. Seven companies (78%) had plans either to begin collecting product usage data or to improve current practices in the future. The key motivation behind these plans was the possibility to assess customer value and enable data-driven decision making. Product usage data was considered *"an excellent tool […] to see in which features to invest [and] how to improve them […]. And also for […] directly guid[ing] our development efforts."*

Despite the wealth of techniques used to collect customer feedback, their use in systematic, continuous experimentation with customers was rare. Experimentation based on explicit, business-driven assumptions only appeared to be an integral development practice in one (startup) company. Four companies (44%) used A/B or multivariate testing, but most only used it occasionally and not necessarily in a systematic way. Additionally, three companies (33%) had plans to begin using A/B testing or to improve current practices. The unsystematic approach to experimentation was also acknowledged by some of the interviewees: *"Whether we are systematic and very good, I have some doubts. It's a little bit ad hoc. So 'Let's have a tagline like this, and maybe like that. Okay, let's put it up there [to production] and let's see'. […] So[…] it is not very thorough and not very scientific."*

The collected customer feedback was typically analyzed to extract work items which were then prioritized into a product backlog. There was some variation in how the interviewees described their approach to feedback processing. Particularly the startup representatives emphasized the need to explore the feedback beyond face value in order to generate new ideas and innovations: *"The interesting thing is their [the customers'] complaint, not the solution that they are providing."*

The level of involvement of different stakeholders in analyzing customer feedback varied: in some cases, both management and the development team were heavily involved with analyzing the feedback and the responsibility was shared. In other cases, the responsibility was on management roles but all the feedback was reviewed together with the team. Lastly, particularly in the larger companies, the process was management-led and the development team mainly based their work on a ready-made product backlog. Some interviewees considered this problematic since it may lead to the loss of valuable insights: *"[T]here is still a lot [of room] for improvement in that area [sharing customer information with the development team]."*

Two divergent approaches emerged regarding the influence of customer feedback on business strategy and goals. First, some company representatives considered that the strategy is continuously being revised based on the feedback. This approach was predominant among the startup companies. As one interviewee said: *"Our strategy is to experiment."* In the second approach, business strategy and goals were considered more stable and therefore not directly influenced by the customer feedback. This approach appeared to be more typical to established companies.

4.4 Challenges with respect to Continuous Experimentation

Fig. 1 gives an overview of the key domain-independent challenges that were identified in this study. Half of the company representatives considered organizational culture a major obstacle to moving towards an experimental mode of operation: *"I would say that the technical things are not [...] even close to the weight of the cultures' obstacles."* Another interviewee agreed that trouble in embracing experimentation *"has nothing to do with technology"*. The overarching issues with respect to organizational culture included a perceived lack of agility, proactivity, and transparency – either within the company or in relation to the company's customers. While cultural challenges were remarked upon by the representatives of both established and startup companies, the general impression was that the more fundamental issues were brought up by the established companies.

Concern over slow release cycles was one of the central themes in terms of product management. Reasons for this perceived sluggishness included R&D task overload and bottlenecks in the development process. Focusing on products and features that create the most customer value was seen as a way to speed up development: *"I don't think you can accelerate anything. What you can do is do less."*

Identifying the metrics to evaluate created customer value and product success was a challenge both in relation to dedicated experiments and to the general observation of product usage. In the words of one interviewee: *"To measure the right thing is the hard thing, to know [...] what is relevant. I think you can easily measure such a lot of things that you [...] lose sight of the forest for all the trees. And then you just optimize irrelevant things."* Particular challenges related to which metrics and techniques of customer feedback collection to use when scaling up a product: *"You can't throw big data analytics on this [product] with a few thousand people, but you can't really [...]interview each [...] one of them [...] either."*

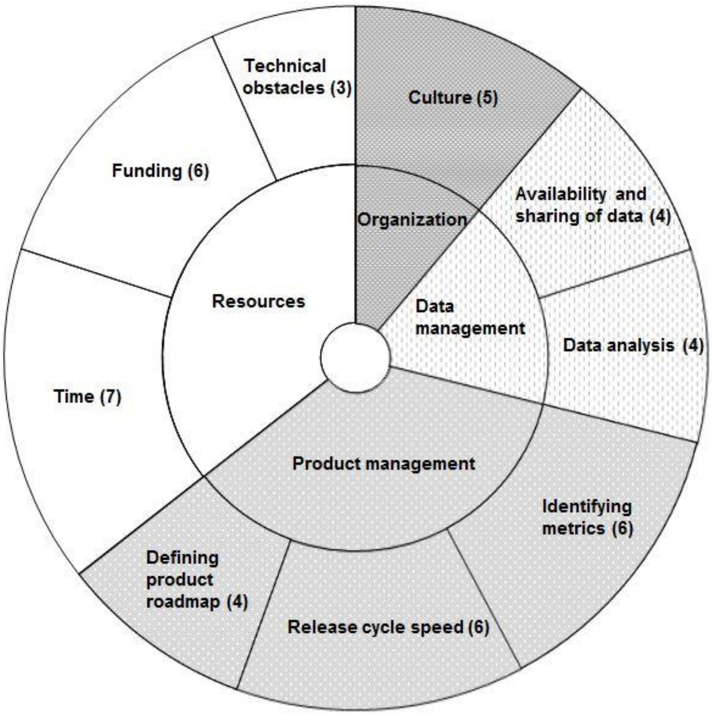

Fig. 1. The key domain-independent challenges with frequency of occurrence by participating company (outer circle) sorted by topic areas (inner circle).

A further set of issues was related to defining the product roadmap. Identifying a truly viable MVP was considered *"very easy to say, very hard to do."* As regards established products, one interviewee described formulating a product backlog as *"black magic"* as it could be so challenging to combine both the product vision and the requests and demands of various customer organizations.

Interviewees also expressed concern over deficiencies in the analysis of collected customer feedback and other data: *"There's too little analysis of available data, we should actually utilize […] the existing data more in our decision making so that the element of gut feeling or some kind of intuition would be minimized."* Lack of time and analytic expertise emerged as possible reasons for inadequate data analysis. Obstacles were also encountered in the availability and sharing of data with all relevant stakeholders. As one interviewee said: *"The data is scattered all over the place. […] [W]e are quite far from providing [a] really convenient, broad spectrum of data to all of the employees."*

Limited resources were a recurrent theme in the interviews. On the other hand, some interviewees emphasized the potential long-term benefits of investing in experimentation. Technical obstacles to experimentation were barely featured in the interviewees' commentaries; there were only three cases in which technical concerns restricted experimentation or had done so in the past. Moreover, these concerns appeared to be primarily linked to insufficient resources rather than insurmountable technical problems.

In addition to the domain-independent issues discussed above, obstacles specific to the B2B domain emerged. Five B2B company representatives considered the customer organizations' culture a challenge to experimentation. For instance, customers were not always able to give feedback or participate in development and experiments: *"[I]t's more like we pull the feedback from them, not that they push it to us. So people are very reluctant […] [t]o give feedback."* Lack of time was the main supposed reason for the customers' disinclination to participate more. A second obstacle, cited by four companies, was limited access to end users. Some interviewees considered that improving product usage data collection would alleviate these challenges. However, customer consent could not be taken for granted: *"[I]t might be difficult to get some of the customers to agree that we can monitor their users and what they do."* This issue was mentioned by three B2B company representatives.

5 Discussion

The study found that the principles of continuous experimentation resonated well within the software industry: there was a wish to focus on customer value creation and data-driven decision making. Many of the contributing companies' current practices supported these aspirations: agile development was prevalent, continuous integration was utilized, and release cycles were reasonably short. Companies were attempting to further shorten release cycles for instance by focusing on key functionalities – a goal which experimentation may help to achieve.

Companies collected a wide range of direct customer feedback, but the collection of implicit customer feedback in the form of product usage data was not ubiquitous and was often hampered by insufficient product instrumentation. However, the potential in product usage data had been acknowledged and most companies had plans to develop their procedures in this respect. These findings are in line with [12,13], suggesting that there is untapped learning potential in product usage data. On a related note, identifying which product metrics to follow and how to analyze the results remained a major challenge.

The present study found experimentation to be systematic and continuous in only one startup company. In addition, several companies expressed interest in A/B testing. This suggests that many practitioners are aware of the possible benefits of embracing an experimental approach to software development. It is also noteworthy that besides controlled experiments, a wide array of customer feedback collection techniques can be used systematically (for examples, see [1], [12]).

The connection between product vision, business strategy, and technological product development is central to continuous experimentation [4] and business alignment [24]. Experiments integrate these aspects by providing empirical data to support both product-level and strategic decision making. This study found a highly flexible approach to business strategy management to only be typical of startups. As Fagerholm et al. [4] note, the continuous experimentation model is derived from a startup environment, and different variants of the model may be required to support other scenarios, possibly in a domain-specific manner.

Innovation is a key feature of a well-functioning experiment system [1]. The present results suggest that the collaboration between the R&D organization, product management, and customers is sometimes insufficient to fully support innovation. There were challenges in sharing and reviewing the collected customer feedback and product usage data with all necessary stakeholders. In particular, the development team was not always involved enough in the process. Furthermore, obtaining relevant customer and end user feedback was often a challenge in B2B environments. These factors may result in innovation potential being lost.

To summarize, it appears that although the majority of companies have not yet reached the stage of continuous experimentation, many are proceeding towards it as outlined by the "Stairway to Heaven" model [3]. Organizational culture has a major role in this transformation. Since an experiment-driven approach to software development is still relatively new [1], companies have had little time to transform their culture and practices accordingly. On the other hand, agile development is a well-established practice, but organizational culture is still cited as the key barrier to further agile adoption, as well as a leading cause of failed agile projects [2]. Similarly, the present study indicates that in many cases, further efforts are required to promote an experimental organizational culture.

Threats to Validity. In accordance with Easterbrook et al. [25], four commonly used criteria for validity are discussed below in the context of this study.

Construct validity was mainly threatened by possible misunderstandings between researchers and interviewees. To diminish this risk, the overall goals of the study and the central concept of continuous experimentation were shared with participants prior to the interviews. Furthermore, the use of semi-structured interviews enabled the asking of clarifying questions for all involved parties. Clarifications were also requested afterwards from the interviewees via email when necessary.

External validity in the sense of statistical generalizability is not the aim of qualitative surveys [15,16]. However, despite the limited scope of the study, a variety of companies represented by interviewees from different roles contributed to it. The authors therefore consider the results to be well grounded in actual practice.

Steps taken to improve the study's reliability included the development and review among the researchers of the interview guide and the analytic codebook. Finally, internal validity, with its focus on causal relationships, was not highly relevant to the present, mainly descriptive study.

6 Conclusions

This paper presented a qualitative survey on companies' experiences of software development as an experiment system. The study found that while many of the current development practices supported experimentation, the state of the practice is not yet mature. Although a broad array of techniques was employed to collect customer feedback, systematic experiments with customers are rare. Moreover, many companies do not use product usage data to learn about customer needs, and product instrumentation is often inadequate. Finally, the collaboration between the R&D organization, product management, and customers sometimes appear insufficient for supporting an innovative, experimental approach.

Key challenges in embracing experimentation are related to transforming organizational culture, achieving sufficiently rapid release cycles, identifying metrics for evaluating customer value and product success, and ensuring that the collected customer and product data is carefully analyzed by relevant stakeholders. Adequate resources also need to be secured. Additional challenges are faced by business-to-business companies.

Acknowledgements. We wish to thank the participants of the study for their time and contributions and the reviewers for their valuable comments. We would also like to thank the Finnish technology agency, Tekes, for funding the Cloud Software Factory project, and the Need for Speed program, under which the proposed study was undertaken. This paper is based on thesis work [26] completed at the University of Helsinki.

References

1. Bosch, J.: Building Products as Innovation Experiment Systems. In: Cusumano, M., Iyer, B., Venkatraman, N. (eds.) Software Business. LNBIP, vol. 114, pp 27–39. Springer, Heidelberg (2012)
2. Version One: The 8th Annual “State of Agile” Survey, `http://www.versionone.com/pdf/2013-state-of-agile-survey.pdf`
3. Holmström Olsson, 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: 38th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), pp. 392–399. IEEE Press (2012)
4. Fagerholm, F., Guinea, A.S., Mäenpää, H., Münch, J.: Building Blocks for Continuous Experimentation. In: 1st International Workshop on Rapid Continuous Software Engineering, pp. 26–35. ACM, New York (2014)
5. Ries, E.: The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Business, New York (2011)
6. Kohavi, R., Deng, A., Frasca, B., Walker, T., Xu, Y., Pohlmann, N.: Online Controlled Experiments at Large Scale. In: 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1168–1176. ACM, New York (2013)
7. Tang, D., Agarwal, A., O'Brien, D., Meyer, M.: Overlapping Experiment Infrastructure: More, Better, Faster Experimentation. In: 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 17–26. ACM, New York (2010)
8. Steiber, A., Alänge, S.: A Corporate System for Continuous Innovation: The Case of Google Inc. European Journal of Innovation Management, 16, 2, pp. 243–264 (2013)
9. Amatriain, X.: Beyond Data: From User Information to Business Value Through Personalized Recommendations and Consumer Science. In: 22nd ACM International Conference on Information and Knowledge Management, pp. 2201–2208. ACM, New York (2013)

10. Adams, R.J., Evans, B., Brandt, J.: Creating Small Products at a Big Company: Adobe's Pipeline Innovation Process. In: CHI ‘13 Extended Abstracts on Human Factors in Computing Systems, pp. 2331–2332. ACM, New York (2013)
11. Davenport, T.H.: How to Design Smart Business Experiments. Harvard Business Review, 87, 2, pp. 68-77 (2009)
12. Holmström Olsson, H., Bosch, J.: Post-deployment Data Collection in Software-Intensive Embedded Products. In: Herzwurm, G., Margaria, T. (eds.) Software Business. From Physical Products to Software Services and Solutions. LNBIP, vol. 150, pp. 79–89. Springer, Heidelberg (2013)
13. Holmström Olsson, H., Bosch, J.: Towards Data-Driven Product Development: A Multiple Case Study on Post-deployment Data Usage in Software-Intensive Embedded Systems. In: Fitzgerald, B., Conboy, K., Power, K., Valerdi, R., Morgan, L., Stol, K. (eds.) Lean Enterprise Software and Systems. LNBIP, vol. 167, pp. 152–164. Springer, Heidelberg (2013)
14. Münch, J., Fagerholm, F., Johnson, P., Pirttilahti, J., Torkkel, J., Järvinen, J.: Creating Minimum Viable Products in Industry-Academia Collaborations. In: Fitzgerald, B., Conboy, K., Power, K., Valerdi, R., Morgan, L., Stol, K. (eds.) Lean Enterprise Software and Systems. LNBIP, vol. 167, pp. 137–151. Springer, Heidelberg (2013)
15. Fink, A.: Analysis of Qualitative Surveys. In: Fink, A. (ed.) The survey handbook, pp. 61–78. SAGE Publications, California (2003)
16. Jansen, H.: The Logic of Qualitative Survey Research and its Position in the Field of Social Research Methods. Forum: Qualitative Social Research, 11, 2, pp. 1–21 (2010)
17. Runeson, P., Höst, M.: Guidelines for Conducting and Reporting Case Study Research. Empirical Software Engineering, 14, 2, pp. 131–164 (2009)
18. Robson, C.: Real World Research: A Resource for Users of Social Research Methods in Applied Settings. Wiley, Chichester (2011)
19. Lindgren, E., Münch, J.: Interview guide and codebook for the paper “Software Development as an Experiment System”, `http://dx.doi.org/10.6084/m9.figshare.1254619`
20. Need for Speed research program (N4S), `http://www.n4s.fi`
21. Vinson, N., Singer, J.: A Practical Guide to Ethical Research Involving Humans. In: Shull, F., Singer, J., Sjøberg, D.I.K (eds.) Guide to Advanced Empirical Software Engineering, pp. 229–256. Springer, London (2008)
22. ATLAS.ti Scientific Software Development GmbH, `http://www.atlasti.com`
23. Ståhl, D., Bosch, J.: Modeling Continuous Integration Practice Differences in Industry Software Development. Journal of Systems and Software, 87, pp. 48–59 (2014)
24. Basili, V., Heidrich, J., Lindvall, M., Münch, J., Regardie, M., Rombach, D., Seaman, C., Trendowicz, A.: GQM+Strategies: A Comprehensive Methodology for Aligning Business Strategies with Software Measurement. In: DASMA Software Metric Congress (MetriKon 2007): Magdeburger Schriften zum Empirischen Software Engineering, pp. 253–266. Kaiserslautern, Germany (2007)
25. Easterbrook, S., Singer, J., Storey, M., Damian, D.: Selecting Empirical Methods for Software Engineering Research. In: Shull, F., Singer, J., Sjøberg, D.I.K (eds.) Guide to Advanced Empirical Software Engineering, pp. 285–311. Springer, London (2008)
26. Lindgren, E., Münch, J. (supervisor), Männistö, T. (supervisor): Exploring Software Development as an Experiment System. Master’s Thesis. University of Helsinki (2015)