UNIVERSITY OF HELSINKI

https://helda.helsinki.fi

Introducing Continuous Experimentation in Large Software-Intensive Product and Service Organizations

Yaman, Sezin Gizem; Munezero, Myriam; Münch, Jürgen; Fagerholm, Fabian; Syd, Ossi ...

2017-11-01
Elsevier Inc.
http://hdl.handle.net/10138/297826
Yaman, S G, Munezero, M, Münch, J, Fagerholm, F, Syd, O, Aaltola, M, Palmu, C & Männistö, T 2017, 'Introducing Continuous Experimentation in Large Software-Intensive Product and Service Organizations', The Journal of Systems and Software, vol. 133, JSS10002, pp. 195-211. https://doi.org/10.1016/j.jss.2017.07.009

Downloaded from Helda, University of Helsinki institutional repository. https://helda.helsinki.fi
This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.
Please cite the original version.

‘Introducing Continuous Experimentation in Large Software-Intensive Product and Service Organizations

Sezin Gizem Yaman[1], Myriam Munezero[1], Jürgen Münch[1,2], Fabian Fagerholm[1], Ossi Syd[3], Mika Aaltola[4], Christina Palmu[4], Tomi Männistö[1]

[1]University of Helsinki,
Department of Computer Science
P.O.Box 68, 00014, University of Helsinki, Finland
{sezin.yaman, myriam.munezero, fabian.fagerholm, tomi.mannisto}@cs.helsinki.fi

[2]Reutlingen University
Danziger Straße 6
71034 Böblingen, Germany
juergen.muench@reutlingen-university.de

[3]Solita Oy
Alvar Aalto Street 5, 00100 Helsinki, Finland
ossi.syd@solita.fi

[4]Oy L M Ericsson Ab
Hirsalantie 11, 02420 Jorvas, Finland
{mika.aaltola, christina.palmu}@ericsson.com

Abstract

Software development in highly dynamic environments imposes high risks to development organizations. One such risk is that the developed software may be of only little or no value to customers, wasting the invested development efforts. Continuous experimentation, as an experiment-driven development approach, may reduce such development risks by iteratively testing product and service assumptions that are critical to the success of the software. Although several experiment-driven development approaches are available, there is little guidance available on how to introduce continuous experimentation into an organization. This article presents a multiple-case study that aims at better understanding the process of introducing continuous experimentation into an organization with an already established development process. The results from the study show that companies are open to adopting such an approach and learning throughout the introduction process. Several benefits were obtained, such as reduced development efforts, deeper customer insights, and better support for development decisions. Challenges included complex stakeholder structures, difficulties in

defining success criteria, and building experimentation skills. Our findings indicate that organizational factors may limit the benefits of experimentation. Moreover, introducing continuous experimentation requires fundamental changes in how companies operate, and a systematic introduction process can increase the chances of a successful start.

Keywords: Continuous experimentation, Experiment-driven software development, Product management, Agile software development, Lean software development, Lean startup

1 Introduction

When companies aim to ensure the success of their products and services, the utilization of data in making development decisions offers a powerful means of finding the right features that provide real value to customers. This requires companies to continuously discover what customers need through direct customer feedback and observation of usage behavior. The findings lead to better knowledge of the customers' requirements and their behavior, which can then be used to guide the development process (Laage-Hellman et al., 2014). In addition, the findings also reduce uncertainty for the development teams. Companies are recognizing the need to transition their traditional research and development (R&D) activities toward experiment-driven systems that support continuous customer feedback and mechanisms to better capitalize on such feedback (Fagerholm et al., 2014).

Using data to support decision-making in software development is not new, and several authors have described a number of approaches and case examples which illustrate what kind of analyses can be made. Large volumes of usage data became available with the rise of web sites and applications, and methods for mining such data have been applied in personalisation, system improvement, site modification, business intelligence, and usage characterisation (Srivastava et al., 2000). Similarly, there are examples of instrumenting software running locally on users' devices and analysing the resulting data to gain insights on, e.g., performance issues (Han, 2012). Pachidi et al. (2014) propose a method to guide the analysis of data collected during software operation, using three different data mining techniques to produce a classification analysis, user profiling, and clickstream analysis to support decision-making.

Whereas data mining can be performed in an exploratory manner without many up-front assumptions, an experiment-driven approach focuses on testing important assumptions about a software product or service. Although the benefits of adopting an experiment-driven approach have been outlined (Olsson et al., 2012; Karvonen et al., 2015; Fagerholm et al., 2014, 2016), the process of starting and methodically doing experiments is unfortunately not very clear. Even though many experiment-driven approaches exist, there is no de facto standard approach and it is not easy to select one and apply it to any company context. While experiment-driven design and software development are current trends, much of the discussion around this topic is high level and practical methods are still poorly understood. As echoed by Fagerholm et al. (2014), conducting experimentation is not an easy task, and companies looking to adopt an experiment-

driven approach to product or service development face many challenges. Adopting such an approach requires companies to have or be willing to create the necessary processes, structures, technological infrastructure and culture to enable experimentation. Coupled with inexperience, existing ways of working, and other factors, moving toward experiment-driven development might seem like a daunting task.

In this article, we present a multiple-case study in which we describe how continuous experimentation was introduced in the product and service development of two large software-intensive companies, one in the digital business consulting and services domain, and the other in the telecommunications domain. The introduction process was led by a team of researchers with prior experience in continuous experimentation. Facilitating experts with an academic background assisted the companies in structuring experiments, provided learning opportunities, and encouraged the companies to think through details they might have skipped due to everyday operative work.

Continuous experimentation takes the principle that product or service ideas can be developed by constantly conducting systematic experiments and collecting user feedback. The term *continuous r*epresents the iteration and sustainability of the approach; however, when introducing the approach, the focus should be on completing the first experimentation cycles methodically which can then be repeated. In this study, we have conducted first experimentation cycles with the case companies with this motivation.

The findings in this paper provide new insights into the activities involved in the introduction process, as well as the relevant decisions made, benefits gained, and challenges faced during the process. In both cases, the companies were able to successfully conduct an experiment and gain the benefits of continuous experimentation early on, such as the ability to make development decisions based on data and not opinions. The study further reveals that starting should not be seen as a hurdle; it can begin at the team level, with teams that include a person with the ability to make development decisions in the company, with small-scale experiments and can later expand to a larger scope.

We do not present the activities for introducing continuous experimentation in companies as prescriptive; rather, we describe a possible set of activities and events involved. At this stage, we do not perform an evaluation of the introduction process, but rather we observe the benefits and challenges as experienced by companies in the software development domain as well as gain a deeper understanding of how the introduction process can be carried out. The findings support existing knowledge and provide new insights that are important for researchers working to develop improved models with clear guidelines that companies can use to easily start.

The rest of the paper is structured as follows. Section 2 presents the background and related work relevant to this study. The study's research questions are presented in Section 3. Section 4 describes the research method, including the case companies, as well as the data collection and analysis approach. Results of the study are described in Section 5, and a full discussion of

the findings is given in Section 6. In Section 7, we discuss the validity of the study. Conclusions and potential future work based on the paper are in Section 8.

2 Background and related work

In this section, we consider the role of experiment-driven development as a means of testing critical product assumptions in the software development process. We take a closer look at how experimentation in software development has emerged in time; followed by existing experimentation models. Here, the term “model” refers to the simplified description that was developed to capture and/or guide the process of experimentation. Coupled with the researchers’ prior work and practical experience in the field, scientific databases were queried with various related keywords such as “experimentation” “continuous experimentation”, “experiment model or framework or method” in the fields of software engineering, computer science, information systems and business information in order to capture as much as related work as possible and identify the existing experimentation models.

Based on these, we identify the common and core elements of the existing experimentation models. These core elements serve as a basis for the researchers to define an initial process for introducing an experiment-driven approach, i.e., continuous experimentation, in the case companies.

2.1 Experimentation in software development

Today’s software development environments are fast-changing, with unpredictable markets, complex and uncertain customer requirements, rapidly advancing information technology, and pressures to deliver products rapidly. To compete and survive in these environments, organizations have to develop, release, and learn from their software products and services quickly (Tichy et al., 2015). Hence, many software companies have adopted or are adopting agile practices, which champion flexibility, efficiency, and speed in developing software (Highsmith and Cockburn, 2001).

Some companies have adopted agile from the start, but many established companies are based on a different approach and operating philosophy. Such companies must undergo a profound transformation if they wish to adopt an agile approach. In the transition path to agile development, there are several steps through which a company must go, as described in the “stairway to heaven” model (Olsson et al., 2012). The model builds on three principles: software is evolved or developed through frequent deployment of new versions, customer data is used throughout the development process, and new ideas are tested with the customers in order to drive the development process and increase customer satisfaction. The final stage in this evolution path, which is termed “R&D as an experiment system,” is the stage at which the whole R&D system is driven by real-time customer feedback and development is able to respond to customers’ present needs. At this stage, deployment of software is seen not as the delivery of a final product but as a way to start, test, and revise functionality (Olsson et al., 2012). This

requires the ability to build data collection components and the capability to use the collected data effectively (Olsson et al., 2012). Karvonen et al. (2015) extended the model by integrating it with practices that are important for companies evolving toward the final stage.

Reaching this experiment-driven stage of software product and service development promises several benefits and can help organizations fulfill the aim of learning quickly and surviving in today's software development environments. Incorporating experimentation into software development not only allows for quick delivery of value to customers but also helps companies make decisions based on customer or user data rather than on just opinions (Rissanen and Münch, 2015). Through experiments, organizations can be more informed about which features to fully implement, thus helping them avoid developing features or products that are not valuable to customers (Olsson and Bosch, 2015). As Bosch (2012) states, "the faster the organization learns about the customer and the real world operation of the system, the more value it will provide."

In some domains, experiments are easier to conduct because the underlying technical platform readily supports rapid deployment of features and usage data collection. Web-based applications are one such example, where continuous experimentation has been used even at very large scale (e.g. Tang et al, 2010; Steiber and Alänge, 2013; Adams et al., 2013). The key to successful experimentation with software-intensive products and services, however, may not be primarily technical but has to do with the capability to develop relevant experiments that yield valuable, actionable information. Lindgren and Münch (2015) define several criteria for systematic experimentation in software domain. An assumption should be tied to higher level business considerations and they should be transformed into testable hypotheses. An experiment should be designed and run based on a testable hypothesis and as results are analysed, they should again be linked to business considerations. If the results are not as expected, the reasons can be investigated further, otherwise they should aid decision-making.

2.2 Experimentation models

With many organizations moving toward agile development and adopting experiment-driven approaches to rapid delivery of value, several models have been developed that aim at capturing and aiding the experimentation process. Although there are differences in the models[1], noticeably, many of them present an experimentation development cycle that resembles the build-measure-learn (BML) feedback loop, which was codified by Eric Ries and that lies at the core of the Lean Startup approach (Ries, 2011). The BML loop starts by forming one or many falsifiable hypotheses that need to be tested. The build step focuses on creating a so-called minimum viable product (MVP) that has been instrumented for data collection (Ries, 2011). The measure step focuses on using the MVP in a test, thereby collecting data. Once the test has been conducted, the collected data is analyzed in order to validate or invalidate the formed hypotheses. Based on whether the hypotheses are found to be valid or invalid, a

[1] It is not our aim at this point to perform an in-depth analysis of the shortcomings of each model, but rather provide a brief description and identify the common experimentation elements among them.

decision can be made to move the idea to the next stage (i.e., implement a full product or feature), correct the course to test a new fundamental hypothesis, or stop.

2.2.1 Innovation experiment systems model

The innovation experiment systems (IES) model, which forms the final stage of the stairway to heaven model (Bosch, 2012), focuses on innovation and testing ideas with customers to fulfill customer and business needs. The model demonstrates a process that first forms a hypothesis that is typically based on business goals and customer “pains”. Following hypothesis formation, a decision is made regarding the quantitative metrics to be used to test the hypothesis. After this, an MVP or minimum viable feature (MVF) is developed and deployed. The MVP or MVF is then exposed to users or customers for a certain amount of time and the appropriate data is collected. In the final phase, the data is analyzed to validate the hypothesis. If the experiment supports the hypothesis, a full version of the MVP or MVF can be implemented, while if the experiment proved the hypothesis to be wrong, the strategy can be altered based on the implications of the false hypothesis.

2.2.2 Early stage software startup development model

The early stage software startup development (ESSSDM) model also extends the Lean Startup principles and aims at offering more support for operational processes and decision-making for startup companies (Björk et al., 2013). The ESSSDM model in particular provides support for a development team to test multiple product ideas in parallel. However, the main purpose is to identify one product idea that is worth scaling. The model has three parts: idea generation (a collection of problems that need solving), a prioritized idea backlog, and a funnel through which ideas are validated systematically, in parallel, using Ries’ (2011) BML loop. The funnel itself is divided into four stages, each stage consisting of a BML loop. Thus, for each idea, one or many falsifiable hypotheses are formulated and experiments are defined and prepared to test them. Afterward, the experiments are run and data is collected, which is then analyzed. The results and learnings are then documented and, in particular, the learnings are fed back into the business model and the validated learning process, which typically leads to new hypotheses. At the end of each BML iteration, there is an opportunity for the team to reflect upon all that has been learned and to act upon it, with the first decision being typically whether the idea is ready to move on to the next funnel stage.

2.2.3 The RIGHT model

Fagerholm et al. (2016) introduces the RIGHT model for continuous experimentation, that utilizes the BML loop, with blocks that are repeated over time and are supported by a technical infrastructure. Each BML block structures the activities involved in conducting experiments. The model illustrates that the experiments are derived from the product vision, which is connected to the business strategy. The business strategy consists of many assumptions underlying the steps to create a scalable and sustainable business model for the product. However, some of the assumptions have inherent uncertainties that can be reduced by conducting experiments. An experiment thus reduces development risks. Hypotheses are formed based on the assumptions, and and experiments are designed to the test the hypotheses. Based on the

hypothesis, an MVP or MVF with instrumentation for data collection is implemented and deployed. The experiment is then executed for a duration of time, and data from the MVP/MVF is collected. The collected data is then analyzed to support product-level decision-making.

2.2.4 Hypothesis experiment data-driven development model

The hypothesis experiment data-driven development (HYPEX) model captures the development process that supports companies in running feature experiments (Olsson and Bosch, 2014). As with the other models, the HYPEX model aims to shorten feedback loops and promotes the development of MVFs that are continuously verified with customers. The model illustrates a number of steps. The first step is the generation of features that may potentially bring value to customers. The features are generated based on business goals and the understanding of customer needs. However, not all features may be selected for implementation. The selection of which feature to implement is the next step and includes selecting features for which there is uncertainty in either the feature functionality, implementation alternatives, or feature development. Following the feature selection, the MVF is implemented and instrumented for data collection. The authors of the model recommend starting with the implementation of the most important functionality. Using the collected data, the MVF is analyzed for gaps between the actual behavior of the feature in comparison to the expected behavior. If the analysis results show that the gap is small enough, the MVF is finalized. However, if the gap is significant, then the development team generates hypotheses to explain the gap. Alternatively, the team can decide to abandon the feature entirely if it is found to have no added value. Two main categories of hypotheses can be formed: either that the implemented feature is not adequate for the customer to obtain the full benefits or that alternative implementation of the MVF will yield a different outcome. In the former case, the MVF is extended in order to collect more accurate metrics.

2.2.5 Qualitative/quantitative customer-driven development model

The qualitative/quantitative customer-driven development (QCD) model views requirements as hypotheses that must be continuously validated throughout the development cycle in order to prove customer value rather than being set in stone early on in the development process (Olsson and Bosch, 2015). The hypotheses are normally derived from business strategies, innovation initiatives, qualitative and quantitative customer feedback, and results from ongoing customer validation cycles. If the customer feedback technique used is qualitative, then the validation cycle consists of direct interactions with customers, resulting in smaller amounts of data. If the technique is quantitative, the validation cycle consists of deploying the feature in the product, instrumented for data collection on feature usage, and then storing the data in a product database. The data is then analyzed and used to decide whether to re-prioritize the hypothesis.

2.3 Core elements of experimentation

In the previous sections, we described existing models that aim to guide organizations in conducting experimentation. However, even though some of the models have been validated within a few software companies, it is not clear how organizations select one that works for them

and how they start using them. This is evidenced by the case companies covered in this study, who were aware of the theories behind some of the models but did not know how to practically adopt them. Among the reasons for this is that organizations' contexts differ; for big organizations with several teams, it is not clear how vision or strategy can drive assumptions held about a product or service feature. For instance, in the RIGHT model, the general vision of the product or service is assumed to exist. Furthermore, not all the models outline who should lead or facilitate the process.

Thus, even though the models described here do a good job of capturing the process, the question of how to practically introduce an experiment-driven approach in software companies is still not very clear. However, the models do present common elements of experimentation, which were used as guides by the researchers in introducing experimentation in the case companies. The elements of the models are listed and described in Table 1. Elements common to all five models include object of experimentation, hypothesis, product and/or feature, process, data collection, analysis, and next steps.

Table 1: Elements of experimentation arising from the models described in Sections 2.2.1 - 2.2.5.

Element	**Description**
Object of experimentation (e.g., concepts, ideas, insights, assumptions, uncertainties, features)	The object of experimentation refers to what drives the experimentation. This can be ideas or problems that need solving, uncertainties related to feature usage, assumptions, or concepts.
Short feedback loop (also rapid customer feedback)	All the models advocate a shorter feedback loop in which a product or feature is deployed continuously in order to get feedback quickly from users and to update the product or feature accordingly.
Value (e.g., increase customer satisfaction, save R&D costs)	Creating, delivering, and capturing value from users or customers is a central motivation for conducting experiments in all the models.
Hypothesis	Hypothesis, although a common element, is described differently in the models. For instance, a hypothesis can be derived from business strategies, innovation initiatives, qualitative and quantitative customer feedback, or results from on-going customer validation cycles. In this case, the hypothesis drives the experiment. In the HYPEX model however, the hypothesis is developed to explain the existence of a gap between actual behavior and expected behavior.
MVP, MVF	Refers to the smallest possible part of a product or feature that adds value to a customer.

Experiment (also a test, validation)	Refers to the logic and actual process of running the experiment.
Data collection	All the models include quantitative or qualitative data collection.
Analysis (e.g., data analysis, gap analysis)	Refers to the process of examining the collected data in order to validate the hypothesis or identify gaps between actual and expected usage behavior.
Next steps (includes learnings and decisions)	The final stage in all the models is to act on the learnings gained from the analysis. This can be to either persevere or pivot the business strategy, finalize, extend, pivot or abandon the feature, move the idea to the next stage, pivot, persevere, or put it on hold in favor of a different idea, or reprioritize the hypothesis.

Existing models portray important perspectives into experiment-driven software development. However, the manner in which to introduce the approach is not trivial for organizations. Thus, we aim to examine how to introduce an experiment-driven development approach, i.e., continuous experimentation, into an organization.

3 Research questions

The main objective of the study was to better understand how continuous experimentation can be introduced in large software development companies, looking at the decision points, benefits, and challenges from the perspective of the two case companies. Based on the study objectives, the following research questions were defined:

RQ1. What are the main activities involved in the introducing continuous experimentation in an organization?

RQ2. What are the decision points that are relevant during this process?

RQ3. What are the benefits observed and gained by the process of introducing continuous experimentation in an organization?

RQ4. What are the challenges faced during this introduction process?

The first research question investigates organizational aspects of introducing continuous experimentation in software companies, including the activities and optimum roles involved, while the second research question explores the decision points that can be encountered during the process. The third research question seeks to identify the benefits of starting to conduct experiments continuously. The fourth research question explores the challenges encountered during the process of introducing continuous experimentation into an organization, as well as its influences.

All the research questions are answered by analyzing the data collected throughout the introduction process. The data collection and analysis methods are explained in more detail in Section 4.2. All the research questions are addressed in Section 5 (Results) and they are explicitly answered in Section 6 (Discussion).

4 Research method

This study follows a multiple-case study approach, with the experimentation introduction process as the unit of analysis, and it adopts an interpretive research approach (Walsham, 1995). The two companies used as cases in this study are both involved in the development of software products and services. In particular, this study focuses on one development team within each company that was involved in the experimentation process. Using case studies allowed us to study the process of introducing continuous experimentation in a real business context, helping us to understand how context characteristics influence its adoption and the manner in which it is carried out (Runeson and Höst, 2009). Additionally, the study has elements of action research, in that the researchers were actively involved in the process being studied (Robson, 2011).

4.1 Research context

The study was conducted as a collaboration between researchers at the University of Helsinki and two software product and service development companies, all participating in the Finnish research program Need For Speed[2] (N4S). The program gave the companies an opportunity to better understand the benefits that are expected from an experiment-driven software development approach. Both companies' interests in adopting the approach motivated them to take part in this study. Companies had prior familiarity with the concept of experiment-driven development and have been conducting ad-hoc experiments such as usability tests with focus groups and with think aloud approaches, however, they did not conduct experimentation in a systematic way. Haphazard or ad-hoc experimentation can produce interesting data, but may fail to reveal the reliable and valuable knowledge required to make good decisions. Systematic experimentations requires the ability to identify areas where experiments are needed, would be beneficial, and would be worth the effort. The researchers have prior expertise in conducting experiments and have developed a model for continuous experimentation in software engineering (Fagerholm et al., 2014, 2016; see Section 2 for more details about the model). Basic facts about the case companies are shown in Table 2, and they are described in more detail subsequently.

Table 2: Key demographic facts of case companies. Company sizes are reported according to European Commission Recommendation 2003/361 (European Commission, 2003), which classifies companies according to headcount and turnover as follows. Micro: <10, ⩽2 M€; small: <50, ⩽10 M€; medium: <250, ⩽50 M€ (both criteria must be fulfilled). In addition, we separate large companies, which exceed the criteria for medium company in the EC recommendation, and very large companies, which we define by headcount ⩾5000 and turnover ⩾500 000 M€. We consider the headcount and turnover of the entire

[2] http://www.n4s.fi

business group but contrast the size with the headcount of the organizational unit that participated in this study.

Company	Domain	Company Size	Software Developers in Case Organizational Unit
A	New business and digital services	Large	Approx. 10
B	Communication technology and services	Very large	Approx. 70

4.1.1 Company A

Company A is a digital business consulting and services company that specializes in developing new business and digital services for clients. Its business offering includes software development, consulting, and service design among others. Company A has adopted an agile way of working where cross-functional team collaboration is emphasized and evolutionary and rapid software development is followed. The established way of working as well as close cooperation with the customer facilitated readiness for adopting continuous experimentation and therefore, the company was a good candidate for this study.

Company A provides a service (Service A), that allows companies to monitor and analyze trends relating to their business and their competitors in media sources. Service A includes various tools such as a search tool, email alerts, and a report tool, all of which allow users of the service to quickly react to events, follow the state of their own business and their competitors' business, and respond accordingly.

Company A provides the service to their client, who uses it to provide further value added services to end customers. The client wants to improve Service A for their customers in order to stay competitive and develop new sources of income. Figure 1 summarizes the stakeholders involved. All the stakeholders involved are organizations and the business network is therefore a business-to-business network.

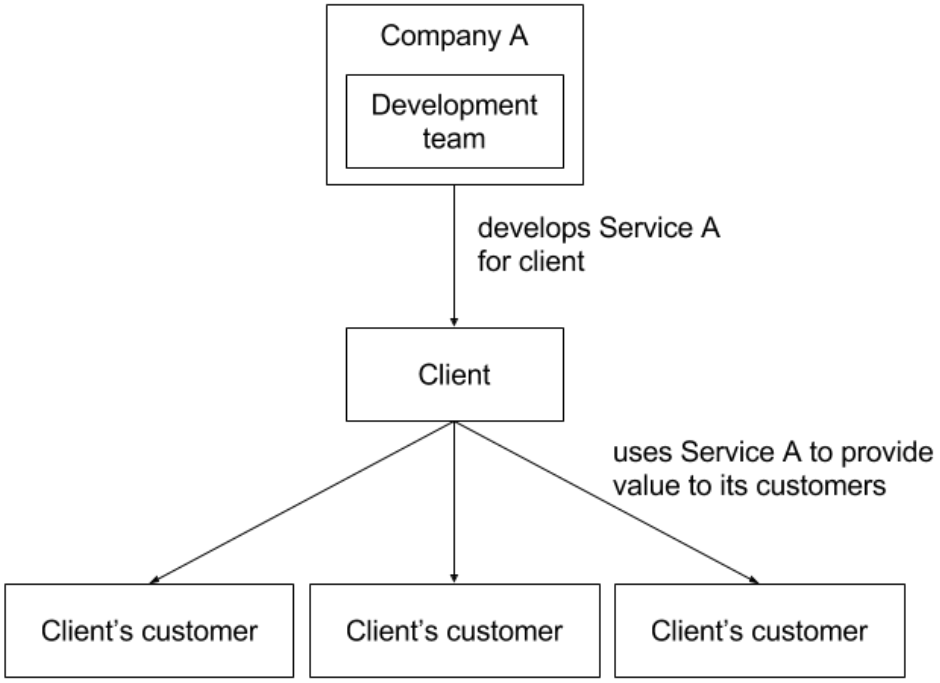


Figure 1: Stakeholders involved with Service A.

Study focus

Evolving the service imposes several risks to Company A. The market for such services as well as the technology involved is highly dynamic. In consequence, customers' needs and potential options for addressing these needs cannot be easily predicted. These needs and solution options need to be understood in order to evolve the service in a successful direction. Continuous experimentation promises to be suited for this task. In addition, Company A's client had limited data about their customers, which was seen as a challenge that could be addressed by introducing the continuous experimentation approach. Fortunately, there was a willingness from Company A to understand the client and its customers better. In addition, accessing the customer usage data was seen as a motivating opportunity to help the development team of Company A prioritize which features to develop and how to evolve the overall service.

The focus of the research study with Company A was on Service A. Within Company A, we report on the work conducted with one of the development teams developing this service. Number of participants are shown in Table 3.

Table 3: Company A participants.

Team	**Team Size**	**People Actively Involved in Collaboration**

Development	4	3 (a business developer, a feature integration manager, and an UX designer)

The roles involved in the collaboration from the company were a business developer (also product owner), feature integration manager and UX designer. In general, all the roles were actively involved in the initial activities of the introduction process, but the business developer and feature integration manager were the most active throughout all the introduction process activities (see Section 6.1)

4.1.2 Company B

Company B is a multinational corporation specializing in providing communication technology and services. The organization is highly distributed, with globally allocated development teams. It operates in the domain of communications technology and provides equipment, software and services to its customers. One of the products the company is developing is a cloud service platform enabling telecom operators to offer connectivity management and billing services to enterprise customers through operator and troubleshooter users. Similarly to Company A, Company B has multiple layers of stakeholders involved with the product, as shown in Figure 2. We have previously described the details of the experiment conducted with this company in Yaman et al. (2016).

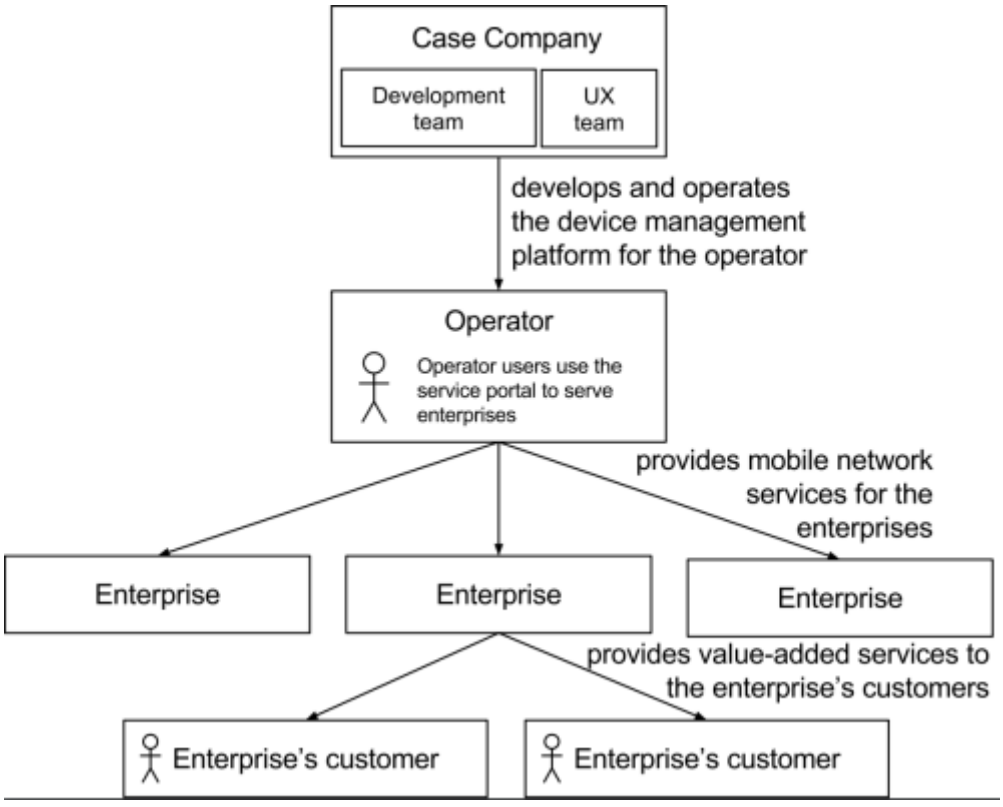


Figure 2: Stakeholders involved with the cloud service platform. (Adapted from Yaman et al., 2016.)

Study focus

The cloud service platform was at the center of the introduction process. More specifically, we focused on one feature of the cloud service platform that was being implemented by a Finland-based team, an activity log. The activity log provides a graphical view of enterprises' subscription communication information. The activity log can be viewed and interacted with by the operators in order to provide troubleshooting service. It is a part of the aforementioned platform, which is under continuous development. The development involves 9 to 11 teams (around 70 people) who are distributed over multiple locations. One of the reasons for focusing on the activity log was that development on it had not yet started and there was a lot of uncertainty about how to develop it in a way that provides value to the end users. This also made the company to be a good candidate for the case study, as one of uncertainties could be the subject of the first experimentation cycle.

Table 4 shows the number of company participants. There were two teams involved in the collaboration: one development team from the cloud service platform and a UX team supporting the development team. Eight people from both the development and UX teams were involved in the introduction process. However, two persons, a technical coach and UX designer, were most active throughout all the introduction process activities (see Section 6.1).

Table 4: Company B participants.

Team	**Team Size**	**People Actively Involved in Collaboration**
Development	4	3 (a business developer, a feature integration manager, and an UX designer)

The teams in Company B follow a lean software development approach. In particular, the development and UX team work closely together and use different methods and tools such as Kanban boards and product roadmaps for product development.

4.2 Data collection and analysis

The primary data sources used in this study are transcripts of audio recordings of face-to-face meetings, minutes and notes of meetings (both onsite and online, including weekly online status meetings), email communication, and open-ended semi-structured interviews from the time of initiation (Spring 2015 for Company A and Autumn 2015 for Company B) to completion of the case studies. In addition, the researchers were provided with background materials (e.g., PowerPoint slides, product and service description notes, user stories) by the company

representatives to be used in the analysis. Altogether, two meetings and two workshops were held with Company A, and eleven on-site and remote meetings with Company B to exchange information about the companies, their products and services and continuous experimentation. As displayed in Tables 3 and 4, two people from Company A and three from Company B were actively involved in the collaboration. Chapter 5 provides a detailed account of the meetings and the roles. All the communication was done in English and thus the collected data was in English. Participant observation was used and all the researchers participating in the meetings took notes which were compared afterwards to ensure consistency.

A database was kept for each company case, as suggested in case study method literature (e.g. Yin, 2009). All case data, including the meeting recordings, transcriptions, minutes and notes of the meetings, background material, interview design, responses and data analysis, were stored in the database. A number of unstructured interviews were held throughout the case duration as well as a post-case semi-structured interview with the active company participants at the end of the cases.

For the analysis of the collected data we followed an iterative thematic analysis approach (Braun and Clarke, 2006; Robson, 2011). The method allowed us to identify, analyze, and finalize themes within the data. We first composed a list of initial themes that emerged both from the data and from core elements of experimentation identified in the literature (see Section 2.3). Afterwards, the data from each case database were extracted and reviewed by two researchers separately in order to first obtain an overall understanding of each case and assign codes to corresponding data, following an inductive and exploratory approach. Then, both researchers compared the codes from each case with the initial themes and held discussion sessions during which the codes were evolved iteratively in a structured way. After several iterations, the researchers reached a state in which no significant codes or themes emerged.

The results of the analysis are presented on two different levels. First, the case-specific results are given (Sections 5.1 and 5.2), formulated as a narrative that describes the events and activities that occurred in the introduction process of Company A and B. These narratives revealed a list of activities that can be followed during an introduction process, as well as an answer to RQ1. Secondly, a cross-case analysis of both cases was conducted (Section 5.3) using thematic analysis organized along the identified codes from the analysis. Cross-case analysis helped us to accumulate knowledge from both of the cases in order to answer the rest of the research questions on the decision points, benefits, and challenges in the introduction process.

5 Results

In this section, we first present the introduction journeys from the perspective of each company separately. This provides answers to RQ1. We then present a cross-case analysis of the two cases, which gives answers to research questions 2, 3 and 4.

5.1 Company A's journey

Through the N4S project, Company A and researchers at the University of Helsinki collaborated on how to start and conduct experiment-driven software development. This took place in spring 2015. Figure 3 captures the timeline of the introduction process for Company A.



Figure 3: Timeline of introduction process for Company A.

Understanding the context

The introduction process started with understanding Company A and its service; this involved an orientation meeting between four researchers and three members of a development team who is developing the Service A. These members were a business developer – who also served as product owner – a feature integration manager, and a UX designer. Hereafter, we will refer to these collectively as Company A.

During the orientation meeting, the researchers first gave a presentation on continuous experimentation and the RIGHT model (Fagerholm et al., 2016). Company A then presented the development history and the current development activities with regard to Service A including its functionality, scope, development schedule, stakeholders involved, and the business goals. During the meeting, it was explained that Service A was a re-implementation and improvement of an existing system. Company A also gave details regarding the processes they undertook to discover and understand user needs. They realized that there was still a need to collect data about how users use Service A in order to understand what is important to them. The Service A usage data can then be used by the development team to prioritize the development activities with respect to functionality, user interface, and business goals. Thus, at the stage of the orientation meeting, Company A was ready and interested in adopting continuous experimentation to support their planned development efforts. Advantageously, Company A was in a stage where the main technical infrastructure, the back-end, and parts of the user interface of Service A tools were already built and in beta testing with people from the client organization.

An important aspect to capture in the orientation meeting was the role of the different stakeholders and their level of interaction with Service A. Understanding these layers helped identify how value is created. In the meeting, three levels of stakeholders were identified (see Figure 1). A challenge was seen with respect to reaching and observing the end users (i.e., the

client's customers) as they were not involved in the beta testing. Thus, in the orientation meeting it was decided that the client, who is also the beta tester, would serve as proxy for the end user during the initial experiments.

The tools included in Service A were also described in more detail during the orientation meeting. Service A uses a search tool to retrieve users' requested information from various sources. The search results are visible to a user in two ways: (1) through an email alert and (2) through a report tool. At the time of the study, Company A anticipated focusing most of their development efforts on the report tool interface, where they would like to provide various functionality for the user, such as sharing, commenting, and rating of search results. With this focus, Company A would like to drive more users to the report tool.

At the end of the orientation meeting, a few ideas had been suggested for which continuous experimentation could help. However, no specific idea was selected. The selection and designing of the experiment were scheduled for the next meeting, which was held as a workshop.

Identifying experiment target

The same participants from the orientation meeting attended the workshop, with the addition of one more researcher. The workshop was structured as a brainstorming session to identify experimentation target ideas. The workshop participants were divided into small groups of three, and each group formulated proposals for what to experiment on.

After each group was finished, all ideas within the groups were presented. The ideas were written down and each was discussed in order to assess the value and feasibility of setting up an appropriate experiment. In this process, those ideas that were obviously too large to be completed in a reasonable time-frame were removed.

During the discussion of the ideas, it became apparent that an important touchpoint for users of Service A was an email alert generated by the search tool and sent to the users. In the earlier implementation of Service A, the email alert showed results linked directly to the original media source. In other words, users had to log into the report tool separately if they wished to see results in the report tool. However, that implementation might result in users only interacting with the email alert and forgetting to log into the report tool, especially if they are satisfied with the email alert. For these users, the interaction with Service A ends with the email alert. But Company A would like to avoid that situation as it is developing the report tool with wide functionality and expects a big user base. Thus, understanding the usage of the email alert and report tool better and ensuring high usage of the report tool, became the target of the experiment.

Running the experiment

After identifying the experiment target, a second workshop was held the week after with the same participants to prepare the experiment design. During the workshop, the assumption about the email alert and report tool was identified. Based on this, a hypothesis was derived and

a plan for testing it was drawn up. Table 5 gives a summary of the experiment design along the common elements of experimentation identified in Section 2.3. At the end of the second workshop, the experiment design was finalized along with action plans and responsibilities for both Company A and the researchers. The researchers were responsible for revising the experiment design, while Company A was responsible for implementing and deploying a MVF. The MVF in this case was an email alert to be sent to the users that included elements linked to the report tool and records when and where a user clicked in the email alert.

Table 5: Experiment design.

Assumptions	Sending email alerts will help drive users more often to the report tool.
Hypothesis	We believe that sending users an email alert linked to the report tool will result in the conversion of users to the tool by 90%. In order to validate this, we will run a two week experiment where users will receive the report tool linked email alerts and we will measure the users that come to the report tool through these alerts.
MVF	Email alert that includes links to the report tool and records users' clicks.
Test subjects	Ten Service A beta testers.
Collected data	Timestamp and location of user clicks for each email alert. Number of email alerts sent to user. Number of times user entered report tool through email alert.
Duration of running experiment	Two weeks.
Data analysis	For each user, calculate ratio of 'number of email alerts that brought the user to the report tool' to the 'total number of email alerts sent to user' in the two-week period.

The experiment was planned to start two weeks after the second workshop, but because of holidays, the process was delayed as the test subjects were then not available. After the holidays, a face-to-face meeting was held between the researchers and Company A with the purpose of enabling the start of the experiment. The experiment ran for the planned duration of two weeks, during which the data was collected. The data was analyzed by the researchers and the results were shared with Company A. Qualitative interviews with the users were also planned, but they were not possible due to time constraints and schedules. This posed a challenge as it would have been beneficial to understand the reasons behind the observed data. For instance, the main analysis result showed that on average, 20% of the email alerts resulted in users entering the report tool, and there were large variations between users, but the reasons remained unexplored.

After the analysis was complete, an evaluation meeting was scheduled to look at the data analysis results, interpret them, and use them as input for decision making. As this was the first experiment, only preliminary results were obtained and lower validity was acceptable. The lack of prior email alert and report tool usage data for comparison is a challenge faced with this kind of experiment, which made it difficult to specify a success criteria in the hypothesis. Tentatively, the criterion was set at 90%, but it was known that this described an ideal situation. It was expected that with further, continuous experiment rounds, a better baseline could be empirically motivated and an improvement target set on realistic grounds.

Although this was a relatively long introduction process for Company A, the company participants stated that the collaboration with researchers was needed as having a structured and academic context ensured that the company participants thought through all the details, which probably would have been skipped in everyday operative work. Company A reported that the described introduction process was a good opportunity for them to learn and develop the actual method and get hands-on experience. It was also stated that in the future, the whole process of planning, designing and running an experiment should be shortened to allow for quicker customer feedback.

5.2 Company B's Journey

The introduction process for Company B took place in autumn of 2015 (see Figure 4). Similarly to Company A, through the aforementioned N4S project, Company B was also aware of the continuous experimentation approach and the expertise of the University of Helsinki team in this area.

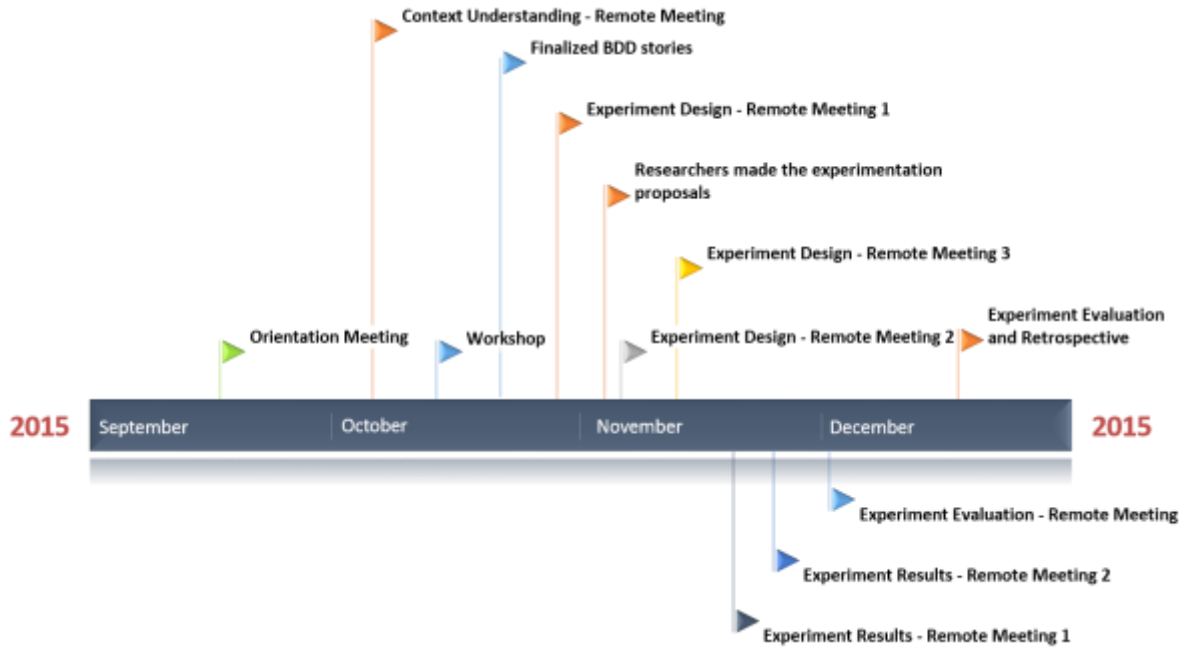


Figure 4: Company B case-study timeline.

Understanding the context

The introduction process officially started with an orientation meeting between Company B and the researchers. The participants of the meeting included three development teams who were developing the cloud service platform (see Section 4.1.2), two people from the UX team, two product owners, two technical coaches, and five researchers.

In the orientation meeting, each of the development teams gave a description of the part of the service on which they were working, the functionality, stakeholders involved, underlying challenges they had faced, and the business goals. Following that, the researchers gave a presentation on continuous experimentation including examples of how it can support the development activities. During the meeting, the participants were divided into three groups, with one development team per group. Within the small groups, the researchers encouraged the teams to think about the service and identify areas in which they were facing problems or uncertainties and what addressing these would mean for their teams as well as for the business. This group work helped the teams take a retrospective look at the goals of their products and consider where experimentation might be needed.

After the orientation meeting, one development team together with the UX team, which we will refer to hereafter as Company B, volunteered to be the starting teams for practicing continuous experimentation. The idea was to start with smaller teams to learn the process and be examples and champions for other teams. Both teams were particularly interested in incorporating experimentation in a more structured and well-thought-out way. In particular, as the development team was a relatively young team in the company, they wanted to combine their work practices with continuous experimentation and improve their communication and customer feedback channels. The UX team especially wanted to reach a stage where they could make decisions supported by data rather than just opinions. The UX team expected that appropriate data would make it easier to communicate and achieve buy-in from the development teams and stakeholders.

An additional online meeting and a workshop were held with the development and the UX teams to gain a deeper understanding of the cloud service platform. In the meeting and workshop, the teams presented the information they had collected about users, including personas, user stories, and behavior-driven development (BDD) stories (North, 2006). In particular, as the platform is continuously evolving into a more robust and modern system, they wanted to eliminate those features that were not being used and improve those features that were frequently used. At the time of the online meeting, one feature of the cloud service platform used to accomplish troubleshooting tasks, called *activity log*, was prioritized for the next release. The activity log provides information about mobile subscription events, such as when a SIM card is registered on the network, a data transfer occurs, or an SMS is sent. The activity log is used by operator users to troubleshoot problems with enterprise subscriptions. A typical scenario would involve troubleshooting during a support call.

Identifying experiment target

The activity log thus became the focal point for an experiment. Reasons for focusing the experiment on the activity log included the following: it was a feature prioritized by the operator; there were uncertainties on how to design the GUI and which functionalities it should include; due to the already planned release date, there was time pressure to make the right decisions expeditiously, with the flexibility that the log could be improved later on; and development of the activity log was just beginning.

In the week following the workshop, Company B prepared and compiled new BDD stories related to the activity log and shared them with the researchers. In turn, the researchers worked with the BDD stories to derive possible experiment design proposals which were then discussed with Company B in a following meeting. Each proposal included a hypothesis and experiment design. From these proposals, Company B then made a selection and informed the researchers. The aim of the selected experiment was to select the right GUI element on the activity log that would best help users complete a troubleshooting task. The main reasons behind the selection were that the GUI element was still in development and there were still uncertainties about its design.

Company B took full charge of completing the experiment design, i.e., refining the hypothesis, specifying the test subjects, and defining the duration of the experiment and how the data would be collected and analyzed. At this point, the researchers only advised Company B regarding the best means to collect data and ways to avoid potential bias during the experiment and analysis of the data.

Running the experiment

In the following week, Company B did a pilot test run of the experiment design with the product owner and based on the feedback, in particular, it was understood that the background information given was not clear. They updated the design accordingly and ran the experiment. Unfortunately, due to the close release deadline, there was not enough time to contact and run the experiment with the real users. Thus, the experiment was run with internal company users that were invited to participate based on availability.

Company B ran the outlined experiment twice. In the first run, Company B quickly realized that there was a flaw in the GUI element versions being compared. Moreover, based on discussions with the researchers, the experimenters (Company B) became aware that they might have introduced some bias into the observation sessions. Thus, Company B determined that the collected data would not be valid for decision making and decided to do a second run. However, even though the first run was not a success, Company B acquired some experience in running experiments and even found an unexpected flaw in their other activity log GUI, which they were able to fix.

In the second run of the experiment, Company B refined the hypotheses to make it more clear (shown in Table 6), defined more distinct experimenter roles, and updated the experiment

design based on the learnings from the first run, which included making the GUI element versions more clear and distinct, as well as increasing the number of versions from five to seven. Consequently, the data obtained in the second run was more valid and reliable. It was also supplemented with test subject interviews. Based on the collected data, Company B was then able to make a decision on the best GUI scenario to implement in the activity log.

Table 6: Experiment details for the second run (Adapted from Yaman et al., 2016.)

BDD story	As an Activity Log user, I want to flush network memory for a subscription so that I can be sure that there is no mismatched information and next I can see when the device connects to the network.
Assumption	After clicking the reconnect button, user will know what is happening and what to do afterwards.
Hypothesis	We believe that with the right feedback message, users are able to tell: (1) what the next action to take is, (2) what the state of device connection is, and (3) what to do if the device does not connect to the network. In order to validate this, users will be shown a set of feedback messages, one at a time, and will be asked to provide answers to the above three criteria. The message with the most "yes" answers for each criterion, especially criterion 1, will be the best message and will be selected.
MVF	Seven mockups (PowerPoint) with different feedback messages.
Test subjects	Seven internal company employees invited by the experimenters based on availability.
Collected data	Yes or no scores for each test subject according to each hypothesis criterion as well as experimenters' observations of test subjects during the experiment and unstructured interview notes.
Duration of running experiment	120 minutes.
Data analysis	Experimenter judgement yes or no scores on each criterion for each feedback message candidate were summed. The sums were used to rank the feedback messages to identify the best message.

After the experiment was complete, an evaluation and retrospective meeting was held where a number of the team members and UX representatives, the new business developer, and the researchers were present. In the meeting, Company B presented and evaluated their three-month journey toward experimentation-driven development.

Overall, even though the introduction process was time-intensive, with several meetings and workshops, Company B was satisfied with the way in which experimentation was introduced. In the last meeting, the participants from the company mentioned that the learnings they got were

worth it and that they expect to carry out experimentation in a more structured way in the future: "*We have not done any structured experimentation before. Now we have the structure"* (Team leader). In the last meeting, the team leader also stated that for their next steps, they plan to present the implemented experiment and learnings to the other teams involved in development of the cloud service and spread the experimentation culture within other development teams as well.

5.3 Cross-case analysis

The cross-case analysis is based on the collected and analyzed data, and the results from companies' journeys as described in Sections 5.1 and 5.2. The main purpose of this section is to aggregate knowledge on the introduction process by identifying commonalities and differences. The evidence from both cases is summarized under three broad categories: (1) decisions with respect to starting, designing, conducting and analysing the experiments, (2) benefits that could be observed or were perceived, and (3) challenges that were faced during this process. Under each category, the common themes are described in tabular form, while differences are discussed afterwards. The themes under the relevant decisions, benefits and challenges categories are derived directly from thematic analysis of the collection data (see Section 4.2). Further analysis of these findings against the existing literature is presented in Section 6.

Decision points

Table 7 presents the relevant decisions made in common by both companies throughout the whole process. Each row contains a decision theme along with a description of the corresponding evidence.

Table 7: Relevant decision points taken.

Relevant Decisions	**Description**
Start adopting experiment-driven approach	Both Company A and B made this decision, which was mainly driven by the following: - To support development decision-making with data rather than opinions or assumptions; - To reduce uncertainty in user requirements; - To gain a deeper understanding of usage behavior with their product and service; - To utilize the convenience of the development stage, i.e., an appropriate infrastructure with access to (proxy) users was ready; - To learn how to conduct experiments in a more methodical manner; and - To expand this way of working at the organizational

	level.
Experiment target	In both of the companies, it was observed that there was an interest in focusing the experiments on UI elements. Reasons for this focus included: - Strict schedules – experiments with UI elements are relatively less time-consuming and simpler to design and run; - As a starting point, it is less risky as it does not require big technological changes and UI modification is easy to deploy, change, or remove based on experiment results; and - Both companies were uncertain about which features to prioritize and how best to implement them. Thus they wanted to use continuous experimentation to reduce those uncertainties.
Means of deriving experiments	In working with both companies, we found different ways of identifying and prioritizing experiments. For instance, in Company A, clarifying the stakeholders' roles and the ways value is delivered was beneficial in identifying relevant assumptions that need to be tested. With Company B, the BDD stories were beneficial in identifying what the user wants to achieve with the product.
Selecting the assumption to experiment on	When working with both companies, we observed that there are certain aspects that companies need to take into consideration when selecting which experiment to conduct. These aspects included: availability of resources such as time and team's availability, current technologies being used, status of current development activities, and availability of test subjects.
Updating the experiment design	As one of the main purposes of doing experiments is being able to make decisions supported by data, it is important for a company to realize when an experiment design has flaws. Both companies iteratively updated their experiment designs. In Company A, the researchers realized that more data might be needed for decision-making and suggested updating the experiment plan and running it again. Company B realized that what was being tested (i.e., GUI elements) was flawed and decided to update and run the experiment again.

The decision to start experimenting was very important, as it determined whether the company or team was willing to invest time and effort in the process. Both companies were motivated to start continuous experimentation in order to reduce uncertainty in their product and service development and gain a deeper understanding of their user and business needs. Despite the common themed decisions, there were differences in the contextual factors that prompted the decisions such as the way of working and the conditions that influenced the selection of the

experiment target, as well as differences in the goals for starting. Company A, for example, was focused on eliminating uncertainties in the development process, while Company B also wanted to improve the communication with all the stakeholders.

Benefits

Table 8 presents the benefits that both company cases experienced throughout the introduction process. Each row represents a benefit along with a description of the corresponding evidence.

Table 8: Benefits gained.

Benefits	**Description**
New insights with respect to business goals and customers	The whole introduction process resulted in new insights for both companies. Especially, with respect to their own business, products and services, and their customers. Such an understanding helped the companies to select and prioritize features that bring value to the business and customers. For instance, Company B stated that through experimentation, they were able to avoid releasing a feature with no or negative value to customers.
Decisions supported by data	By running experiments, it was possible for both companies to make development decisions supported by data rather than by assumptions. For example, through evaluation of the results, Company A was able to evaluate whether to keep, change, or abandon the use of the email alert feature in its current form. Company B was also able to select the best GUI element based on the collected data.
Reduced development effort	Adopting an experiment-driven approach helped companies start reducing their development effort. For instance, Company A was able to see how Service A was being used, allowing unnecessary development efforts to be cancelled for those features that were not creating value for the user or the business. In the case of Company B, they were able to select the right version of the feature to implement without any coding effort.
Improved knowledge on systematic experimentation	Even though both companies were familiar with the concept of experiment-driven development, none of them had conducted experiments in a structured or methodical way before. Toward the end of the journey, both companies stated that the whole process with the researchers improved their knowledge and experience, which can be seen in their confidence re-running the experiments (e.g., Company B). Both companies also stated their intentions doing experimentation continuously;

	Company B was already in the middle of its second experiment at the time of writing of this paper.
Retrospective look	The time taken to review, evaluate, and discuss the business, product, and service, helped the development teams gain new insights on improving their communication and helped them better understand their way of working. For instance, during this process Company B realized that there is a need to improve the communication between the development and UX teams, and conducting experimentation with both teams provided a way to achieve this.

As highlighted in Table 8, the introduction of continuous experimentation offered various benefits for companies, from improved understanding of the value of their services and users, to reduced development efforts. Particularly for Company B, continuous experimentation was also observed to improve communication in their own development teams and give them a retrospective on their ways of working. Furthermore, the introduction process, guided by the researchers, helped the companies to gain the ability to design, implement, and conduct experiments in a more structured and methodical manner. With continued effort, the teams will be able to conduct structured experiments at different levels of their business.

Challenges

Table 9 presents the challenges that were faced by both companies during the process. Each row shows a common challenge next to the description of the corresponding evidence.

Table 9: Challenges faced.

Challenge	**Description**
Access to end users	In both of the companies, reaching end users to be involved in the experiment was found to be challenging. In the case of Company A, it was not possible to reach the users for the qualitative analysis part of the experiment. In the case of Company B, it was difficult to reach end users for conducting the experiment. Alternative ways, such as proxy users, were used to overcome the challenge.
Different stakeholders' have varying values	There was a multi-layered structure of stakeholders in both cases. Although this is expected, especially in large software organizations and multi-sided business models, it was a challenge to isolate and identify which stakeholders should be involved in the experiments and whose value the experiments should fulfil. Discussion sessions, working with stakeholder analysis, and BDD stories helped guide this process.

Experimentation on an evolving product and service	Both of the companies were in the middle of developing a product and service that was evolving from an old solution to a new, modern solution. Thus, when selecting the first experiment to run, care had to be taken not to create any conflicts between completed, current, and planned development, which can take some time and effort to coordinate.
Inexperience with experimentation	Although the aim of the introduction process was to give knowledge to the companies, the fact that they had not conducted structured experiments before resulted in some hesitancy and indecisiveness in attempting to make experiment-related decisions. For instance, in the case of company B, one of the reasons for not involving real users in the experiment was due to lack of confidence and wanting to wait until more experience had been gained.
Forming the hypothesis success criteria	In both cases it was difficult to access both earlier implementations of the companies' services. This prevented the teams from forming hypotheses based on comparison or base data. Also, lack of such data made defining success criteria challenging as there was no comparison point.
Existing and short deadlines	Related to the third challenge of experimenting on an evolving product or service, when a company starts doing experimentation, they have to work with or around existing deadlines and commitments which naturally affect experiment prioritization. This was observed in both Company A and Company B.
Length of the process	The introduction process took a relatively long time as it was a study that had elements of action research. For instance, the process from understanding context to running first structured experiment took 11 months for Company A and 3 months for Company B. This unfortunately might affect the motivation and persistence of the company participants.

Some of the challenges described above are due to the multi-layered structure of the product or service. Such a structure, which was observed in both companies, led to difficulties in identifying each stakeholder's needs and accessing end users for running the first experiments, leading to inability to incorporate end users' feedback in the development process. While it was possible to get useful data for decision making through internal users, as in Company B, this might unfortunately, limit the validity of the experiment results.

Another challenge in conducting the experiments was validating the hypothesis, i.e., defining success criteria. In the case of Company A, we decided with the company to set it at 90% click-through rate on the emails, but availability of previous data regarding report usage would have allowed us to set a more accurate success criterion. Furthermore, the volatile environment in the business area meant that the interest towards the experiment would decrease quickly with

new priorities. Thus obtaining proper baseline data quickly enough proved challenging. There was a similar challenge in Company B, where some experimentation ideas would have required comparative data to validate that a feature in a new software version is better than an earlier version. This was the company's' first structured experiment, and we observed that inexperience with the approach can lead to uncertainty, introduction of biases in experiments, and indecisiveness. In general, researchers' guidance and encouragement helped to mitigate these challenges.

6 Discussion

Our study found that introducing continuous experimentation requires a careful presentation of practices. In our study the introduction process was facilitated by researchers who also served as experts on experimentation. The findings are in line with those of Olsson et al. (2012), who emphasized the need for knowledgeable people to be involved, to take on the role of facilitator and provide guidelines on how to apply experimentation in the development of their products and services. The need for guidelines was also noted by Fagerholm et al. (2014, 2016) in their development of the RIGHT model. Even though many of the benefits and challenges of experimentation that are described in the literature were found to be inline with what we have observed in our study, we particularly identified those that were related to conducting first structured experiments. In addition, we have outlined the activities and the decisions points involved in the introduction process as guidelines, thus contributing novel results to the field as the related work did not provide such knowledge.

The next subsections discuss each of the four research questions outlined in Section 3 and compares the findings with our expectations and related works.

6.1 Activities involved in the introduction process

Following from RQ1, it was identified that the introduction process followed a similar set of activities for both companies, with slight variations depending on the context of the company. The starting point was the willingness of the company to adopt or at least try to integrate the approach in their development process. Adopting experimentation-driven practices for strategic decision-making is one of the key practices identified by Karvonen et al. (2015) for companies to reach the final stage of the "stairway to heaven" model, i.e., R&D as an innovation experiment system. It is critical that companies see the need and are motivated because time and effort need to be invested in the process. The development teams involved in this study did not have prior knowledge about performing structured experiments, which created a need for experts. In this study, the expertise was provided by the researchers from the University of Helsinki.

In our study it was helpful that we started with motivated teams who understood the need for the new development approach. This is in line with Dougherty and Hardy's (1996) findings on innovation. We found that it is also important to select or identify those individuals in the development team who will be active in the introduction process. These become the new experts for future experiments in the companies. In Company A, the driving force was a service

business developer and in Company B it was a product team leader. Similar experiences have been described by Olsson et al. (2012), who recommends starting the transformation with small development teams.

The next activity in our study after establishing motivation was the selection of a product or service (see Figures 5 and 6). This is accompanied by gaining deeper understanding of the company and its product/service context. We observed that in Company A, we commenced with the broad picture of Service A and narrowed down on the critical product features by prioritization in a structured way. In Company B, we started with a specific product feature and investigated the link from there to the higher-level product vision in an exploratory way. Both approaches were successful because they fit the profiles of each company. Understanding the context involved several face-to-face and remote meetings, as well as workshops. Not only were these meetings beneficial for the researchers in understanding the company, but the activities also helped the companies to better understand their own context and work. Notably, including multidisciplinary roles, especially in the beginning of the introduction process, helped in evaluating the necessity for experimentation from different perspectives. For instance, business people provided their insights on product and service features, which were discussed as potential foci for experimentation.

Better understanding of the context and product/service allowed the companies and researchers to identify uncertainties that might pose critical risks to the product development success (e.g., customer risks, problem risks, solution risks). These uncertainties were described as assumptions, which were then formulated as testable hypotheses. Afterwards, these hypotheses were prioritized and appropriate experiments were defined to test the prioritized hypotheses. As echoed by some of the experimentation models covered in Section 2 (i.e., RIGHT, ESSSDM, HYPEX, QCD), all the identified assumptions should be put in a backlog and from there assumptions should be prioritized in order to determine which experiment to conduct next. Prioritization depends on the context, time constraints, and development effort.

In our study, we focused initially on experiments on UI features. This was because they were seen as smaller, and easier to implement, and they allowed feedback to be obtained quickly, which was important because both companies had pre-existing deadlines and the experiment had to be completed within a reasonable time frame. Moreover, it is easier for the researchers to transfer their knowledge and practices using small experiments. This is also in line with Olsson et al.'s (2012) recommendation that it is easier to focus on features than components when shifting to agile practices.

Following the selection of the experiment, the next activity was designing the experiment. Here the researchers used their expertise and experience to facilitate the process, for instance, helping the companies formulate hypotheses from the identified assumptions. The hypothesis should be clear and testable, i.e., it should state the data to be collected, for how long, and what is to be measured. The experiment should be capable of validating the hypothesis. If an experiment is poorly designed, making a decision based on the experimental outcomes will be almost impossible and it should be re-designed and run again. Through hands-on work and

close collaboration through the whole experiment design process, we wanted the companies to learn the process of making decisions driven by data as well as how to establish a short customer feedback cycle. Both aspects are key practices identified by Karvonen et al. (2015) as important practices to have in an innovation system. However, in this study the quick customer feedback could not be established due to the multi-layered structure of the companies.

The last activity that has to be performed by companies is to decide what to do with the results of the experiments (see Figures 5 and 6). The experiments provide information, but ultimately the companies incorporate other factors in making a decision on whether to rerun the experiment, develop the MVF, or abandon the feature.

In summary, the activities followed were as follows: establish need and motivation, identify a development team to start with, understand the context, identify champions within the development team, identify a product or service, identify assumptions, select and prioritize assumptions, draft experimentation design, conduct the experiment, collect and share findings, and make decisions based on the findings. These activities are illustrated with each company's timeline in Figures 5 and 6.

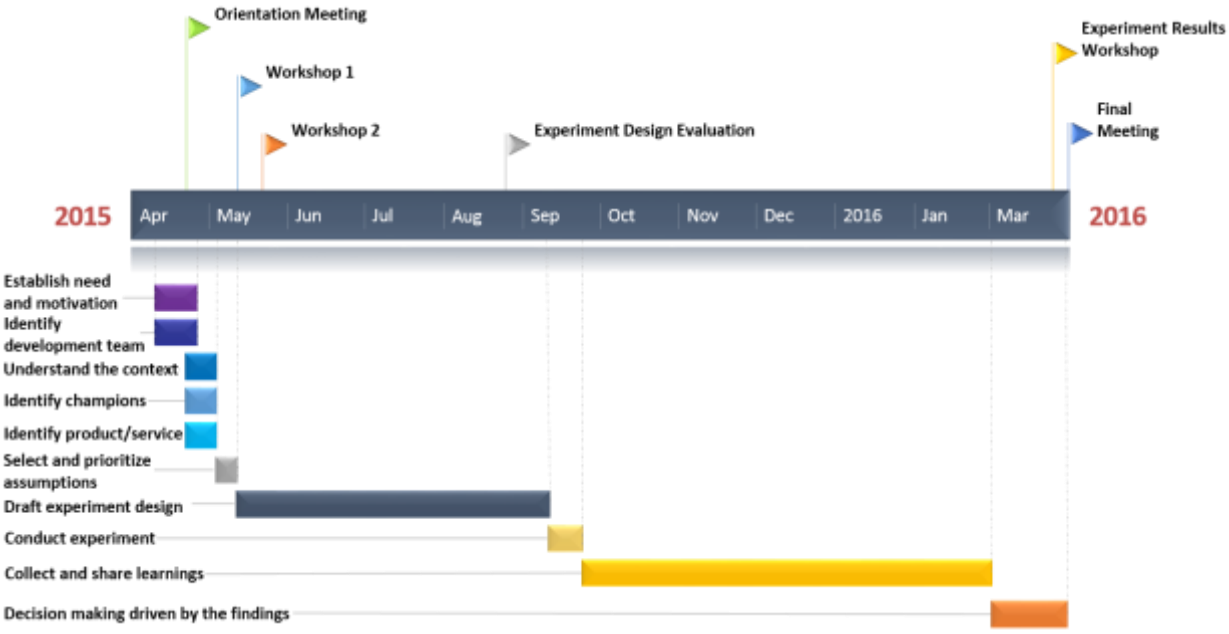


Figure 5: Activities involved during the introduction experiment process for Company A.

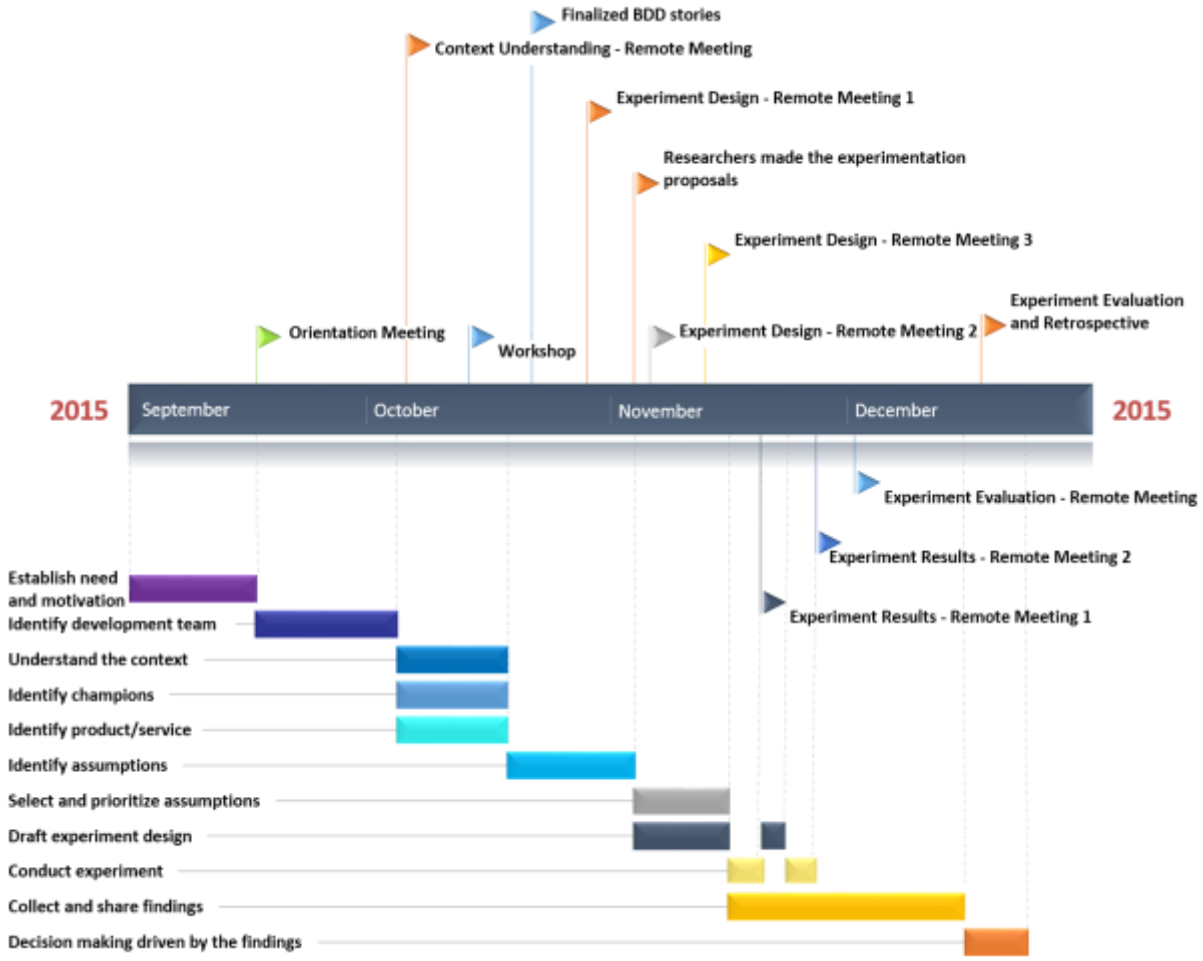


Figure 6: Activities involved during the introduction experiment process for Company B.

In looking at Figure 5 and 6, we see that although the activities involved in the introduction process are similar for both companies, differences in duration and sequencing of the activities exist. For example, for Company A, the initial six activities before drafting the experiment design were performed relatively quickly and in parallel. This was because many of these activities were done during workshops (see Section 5.1). While with Company B, initial activities were more sequential with one or more activities ending before others begin. This could have been a result of the company's structure and way of working, i.e. they needed time in order to prepare for the activities, and consult on the decision points. Another visible difference between Figure 5 and 6 is that while Company A activities are linear, Company B has an iterative set of activities, i.e., draft experiment design and conduct experiment. This was because Company B noticed a flaw in the experiment design and decided to rerun it.

From both figures, we also see that the activity of collecting and sharing findings takes the longest time in comparison to other activities. The activity however is dependent on the number of test subjects, the scale of the MVF and duration of data collection. In particular for Company A, we see that they had two long sets of activities, i.e., draft experiment design, and collect and share learnings. The former was long as a result of holidays and the latter as a result of technically implementing the MVF.

Keeping in mind that only two cases are included, we observe that there is no standard sequence or duration of activities. The activities rely on the company, product and service contexts. We further notice that duration of activities has an influence on the motivation of companies in adopting continuous experimentation. For example, at the end of the 3 month introduction process, Company B still had high motivation and persistence that they immediately embarked on a second experiment. While the 11 month process for Company A might have reduced their motivation, resulting in the experiment not being run for a second time, as would have been advisable. Thus, to maintain motivation, and to also make the introduction process more economical for companies, ways of shortening the introduction process would need to be identified and implemented.

6.2 Relevant decision points during introduction process

Through the collaborative process, based on our RQ2, we were able to observe the type of decisions the companies made and when they made them. For instance, for both companies, the decision to start continuous experimentation was driven by similar needs, such as the need to support development and decision-making processes by data rather than by assumptions or opinions.

Furthermore, a decision on which assumptions to test through running experiments had to be made. This decision included taking into account existing deadlines of the evolving products and services, time and resources availability and the possible learnings to be obtained. It was observed that both companies decided to focus on UI elements that were being developed, because UI elements were a lower risk and did not require much resources from the companies. Validation of small parts of features is also supported by Olsson and Bosch (2014) in the HYPEX model. In particular, UI or UX experiments are possible in many organizations, even ones developing security or performance systems where experimentation would not otherwise be possible (Karvonen et al., 2015). Hence, the choice of where and what to experiment on has to fit with the company goals, current schedules and priorities. Ideally, however, companies should aim to test the most value-creating features and design experiments that would actually test the value of features.

Other decisions made were related to conducting the experiment. Decisions here included selecting the means for identifying and prioritizing the experiments (e.g., using BDD stories for Company B and using the product roadmap for Company A), selecting the test subjects, deciding on operational details of the experiment such as when to run the experiment and for what duration, as well as deciding to update and rerun the experiment in order to reach a correct conclusion (Company B). Additionally, we observed that especially in large companies like Company A and B, decisions are rarely based only on experimentation results but also other organizational factors.

When looking at the roles involved in the decision points, we observed that for Company A, the relevant decisions were made by the business manager and were made mostly during the workshops and meetings with the researchers. In Company B, a majority of the relevant

decisions were made internally by the technical coach in consultation with the UX team, and then communicated to the researchers during meetings. Thus, even though both companies had a similar number of participants in the collaboration, we observed different styles of decision-making but we also observed the importance of having at least one company participant in the collaboration that is able to make decisions.

6.3 Benefits gained by introducing continuous experimentation

Relating to RQ3, the process of introducing continuous experimentations revealed several benefits for the companies as described in Section 5.3. Some of the benefits gained, which were also observed in related work (Olsson and Bosch, 2014 and Yaman et al., 2016), include improved understanding of the need for user involvement in the development process and getting rapid feedback. Moreover, the teams gained new insights and understanding regarding their product and services, the users, and their development processes during the introduction. This was for instance expressed by the business developer in Company A: “we see that understanding end-customers via analytics and experimentation is vital key to successful services and business”. While the technical coach in Company B, realized that without doing experimentation, the chances of creating features with zero or negative value was higher. The technical coach further added that “experimentation made it clear to the team that there is no need to debate between opinions as you can quickly test them with an experiment”.

Another benefit gained was a result of the close collaboration between the researchers and the case companies. The collaboration led to improved practical knowledge about conducting experiments in a systematic way. This involves being able to form hypotheses that can be tested, deciding on the data to collect that is necessary for validating the set hypotheses, and the methods to use to collect and analyze the data. As the business developer in Company A affirmed, “we got hands-on experience from actually doing experimentation” and they understand that “practice will make it perfect”, as stated by the technical coach in Company B.

The need for expertise or experience was also noted by the researchers. Situations arose in which both companies required expert advice on experimentation. For example, the development team from Company B asked researchers about the experimenter effect in order to understand how to eliminate such threats in the experiment design. The answers to such requests can fundamentally affect the experiment design, so it is crucial for the answer to be based on expertise. On the other hand, we were aware that as a facilitator (either external or internal to the companies), care should be taken not to take on all the responsibility. It is important to involve the development teams throughout the process, encouraging them to work together and be hands-on, in order to get the experience and for the learnings to be meaningful.

Moreover, as reported by Olsson and Bosch (2014), one of the benefits of introducing continuous experimentation is that it improves communication in organizational units. This was one aspect in which Company B was particularly interested in and have already devised plans to use experiments as a means to connect different development units and improve communication between them.

As the unit of analysis of the current study is the introduction process, we realize that most of the benefits of experimentation which are outlined in literature (e.g., improvement in user satisfaction, revenue increase) have not been actualized by the companies yet. But, based on the benefits already gained and the motivation and persistence of the companies, we see that those benefits can also be fully achieved in the longer term.

6.4 Challenges faced during the introduction process

Despite the benefits experienced, there were various challenges encountered in the introduction process, following from RQ4. Some challenges faced were a result of the multi-layered structure of the companies which resulted in difficulties determining whose value the different service features fulfilled and how. This challenge was also identified by Olsson and Bosch (2014) in their validation of the HYPEX model and by Rissanen and Munch (2015). The multi-layered structure also played a role in being able to access users of the services. This was also realized by Fagerholm et al. (2014): when companies are not selling their products or services directly to end users, the intermediaries might interfere with the possibility of collecting data from experiments with end users. In many existing experimentation models, the customer and the user are considered synonymous but in both of the companies in this study, this was not the case. This additional layer of understanding about whose value was targeted in the experiment and how to access them represented a new dimension as well.

Both Company A and B made the decision to use different kinds of proxy users to overcome challenges arising from lack of access to end users. Company A decided to use beta testers from the client organization as proxy users and run the experiment with them. Company B used internal employees as the proxy users. The learnings of conducting experiments were still gained, but ideally, feedback from real users would be more beneficial.

Particularly for Company B, not only the multilayered structure but inexperience with experimentation made them hesitant on reaching users before gaining some experience. As one member from the UX team explained, “[The whole] idea of experimentation is quite new to our customers so [there are] kind of political reasons why in the first place we did not contact our customers. It was so agile to do it in-house and we did it so fast with our workmates. [...] We wanted to learn about the continuous experimentation approach and it would be easier to practice it in-house in the beginning.”

In addition, because both companies were in the middle of developing an evolving service, there were existing feature prioritizations and deadlines that influenced the selection of the experiment target and scale of the experiments, as well as resource allocation. Together with the multilayered structure, this made it rather difficult to perform experiments that would have rather determined whether the features are at all necessary or suitable for accomplishing given tasks that have already been found to fulfill user needs (Yaman et al. 2016). For Company B, this resulted in the selection of a small-scale UI experiment, even though a more ambitious or bigger experiment would have provided more value to the teams and the user. Although Company A also selected the target of the experiment to be a UI element, the experiment itself

was of a larger scope, involving technical implementation of a MVF, and the experiment results had a larger impact on development decisions. Defining useful experiments was acknowledged by Björk et al. (2013) as not being easy.

As described by many of the experimentation models, a good, testable hypothesis is needed to drive the experiment forward, including identifying the success criteria and data to collect and measure. Defining metrics and what data to collect was similarly found to be challenging in the work of Olsson and Bosch (2014). Due to the evolving nature of the services being developed by both companies, it was difficult to access data from previous implementations. This resulted in difficulty to set a realistic success criterion in the case of Company A. A conversion rate of 90% was selected as the success criterion, but this was not founded on any previous conversion rate data. However, for the next experiment run, the initial results can be used to set more realistic success criteria. Thus, being flexible with the success criteria in the beginning and running the experiment a few times would be a way to overcome this challenge.

Looking at the introduction process itself, long duration was observed to have a negative effect on companies' motivation and persistence, which is understandable due to the cost accrued. For Company A, the process took 11 months – however, this includes nearly two months of waiting due to holidays. Company A also implemented a MVF for data collection, which lengthened the process duration. In case of Company B, the process took around 3 months. However, Company B conducted a smaller UI experiment, used mockups and internal employees, and had champions actively running the experiments, which resulted in a shortened process. As seen in Section 6.1, both companies progressed through a similar set of activities and conducted one experiment. Thus with careful planning of the introduction process activities, the duration of the introduction process can be shortened to suit company needs.

7 Validity

Validity in case study research can be considered in terms of internal validity, construct validity, and generalizability of the results. With regard to internal validity, the main objective of this study was to obtain descriptive knowledge about introducing continuous experimentation in companies and the study does not directly aim at proving causality. The introduction approach revealed by this study is likely to be of interest to anyone who wants to introduce experimentation in an organization.

Construct validity for this study can be discussed from different points of view. Firstly, as researchers with prior experimentation experience, we were familiar with the benefits of experimentation in software development. For instance, when we constructed RQ3 (i.e., benefits of introducing the approach to an organization), we realized the potential threat of allowing our prior knowledge on the benefits of experimentation to influence the current study. However, during the data analysis stage, we were careful to extract and analyze data based only on the cases, focusing on the introduction processes, rather than on the expected or promised benefits of experimentation. Although the process might still be subject to researcher expectancy bias in general, other researchers not involved in the initial data analysis reviewed

the results in an effort to eliminate such a threat. Furthermore, representatives from the case organizations were available throughout the process and were involved in the writing of this paper to ensure that our interpretations did not conflict with their understanding.

Secondly, the term *experimentation* can cause threats to construct validity as experimentation in software development does not necessarily have the same meaning as in scientific experiments. Although there are similarities, there are fundamental differences; in scientific experimentation, hypotheses are derived from theory or existing knowledge, whereas experimentation in software development is about working hypotheses that have more pragmatic utility in business. Therefore, at the beginning of the introduction process and when forming the hypotheses, we emphasized this difference in experimentation. For example, in the first experiment with Company A, a tentative success criterion for the hypothesis was acceptable, which would not be the case in a scientific experiment.

In terms of generalizability, we are interested in whether the results of this study would be applicable to other software development companies wanting to transition toward continuous experimentation. We cannot make strong claims based on two cases, but the similarity of the findings between the two cases is such that it would not be surprising if they were valid for other companies, as well. Another form of generalization of primary interest to the companies is how to disseminate the experimentation-driven culture and learnings within their own organizations. In this study, the researchers had a facilitator role based on their expertise with experimentation. However, for companies transitioning on their own, the level of understanding of experimentation principles needed or the dangers of attempting to introduce experimentation without much understanding remains unknown on the basis of our results. This, unfortunately, poses a potential threat to the generalizability of the results to situations in which different (or no) facilitators are present.

8 Conclusion

Many companies are interested in experiment-driven development. However, not all companies have the experience, knowledge, or time required to learn the existing approaches and adopt them in their development teams. Even though several experimentation approaches have been developed to help companies perform experiment-driven development, there is a lack of clear guidelines on how to actually start. In this study, we have presented a multi-case study describing the journeys of two software companies during the introduction process of continuous experimentation.

In examining the journeys of the case companies, we identified a set of ten activities involved in the introduction process that were similar to both. These activities are however flexible in duration and sequence to suit the company's circumstances and ways of working. In this study, we further identified relevant decision points made by the companies. These ranged from the companies' willingness to adopt continuous experimentation, the participating development teams, to decisions on which experiment to conduct and how to utilize the experiment findings.

Having a company participant that is able to make decisions was an important part for the introduction process.

Moreover, we identified several benefits and challenges companies can face during the introduction process. For instance, benefits such as making development decisions based on empirical data were observed during the study time frame, whereas other benefits, such as gaining new insights with respect to business goals and customers were beginning to be observed, but are expected to be fully realized as the approach gets more adopted into the companies. During the introduction process, particularly for companies without prior experience with systematic experimentation, it was found to be beneficial to have experts facilitating the process. Especially during the selection of an experiment target and drafting experiment design activities. On the other hand, constraints such as the multilayered stakeholder structure, unavailability of prior product and system data, and existing product release deadlines, hindered the type of experiments that could be conducted for the first time and in turn also limited the benefits that could be obtained from continuous experimentation such as getting feedback from real users.

Regardless of the challenges, we observed in this study that by starting with small teams, first running experiments on small-scale features, and working in collaboration with experts guiding the process, resulted in a successful introduction with valuable learnings for the both the companies and the researchers. We have already disseminated the current findings to the companies involved, and we are continuing to work with them on integrating continuous experimentation further in their development processes.

Also, from the researchers' perspectives, the introduction process resulted in improved understanding of continuous experimentation, which was one of the aims of the study. In future work, we expect to introduce the process to more companies, allowing us to develop a roadmap that can easily help software companies adopt continuous experimentation. Domain-specific variants of the roadmap are expected to exist. In addition, we plan to assess in more detail the expected benefits with respect to the time and effort invested in the introduction process. Such information would assist the development teams in their planning as well as achieving buy-in from other teams or company heads. We would also like to examine the role of experts and how much expertise is required at the starting stage in future studies. Another area where more exploration would be beneficial is the finding that since the customer and the end user can be different entities, there are actually different feedback loops that can be shortened. This distinction is not made in the different experimentation models and should be further investigated.

Acknowledgments

The authors wish to thank Solita Oy and Ericsson for access to valuable case data, and the company representatives for their time and effort spent on this study.

This study was supported by the Need for Speed research program of DIMECC, funded by Tekes, the Finnish Funding Agency for Innovation.

References

Adams, Rob, Bradee Evans, and Joel Brandt. “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, 2013.

Björk, Jens, Jens Ljungblad, and Jan Bosch. "Lean Product Development in Early Stage Startups." In *IW-LCSP@ ICSOB*, pp. 19-32. 2013.

Bosch, Jan. "Building products as innovation experiment systems." In *International Conference of Software Business*, pp. 27-39. Springer Berlin Heidelberg, 2012.

Braun, Virginia, and Victoria Clarke. "Using thematic analysis in psychology." *Qualitative research in psychology* 3, no. 2 (2006): 77-101.

Dougherty, Deborah, and Cynthia Hardy. "Sustained product innovation in large, mature organizations: Overcoming innovation-to-organization problems." *Academy of Management Journal* 39, no. 5 (1996): 1120-1153.

European Commission. “Commission Recommendation of 6 May 2003 Concerning the Definition of Micro, Small and Medium-sized Enterprises (2003/361)”. <http://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:32003H0361>, 2003 (Retrieved 2016-10-26).

Fagerholm, Fabian, Alejandro Sanchez Guinea, Hanna Mäenpää, and Jürgen Münch. "Building blocks for continuous experimentation." In *Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering*, pp. 26-35. ACM, 2014.

Fagerholm, Fabian, Alejandro Sanchez Guinea, Hanna Mäenpää, and Jürgen Münch. "The RIGHT model for Continuous Experimentation." *Journal of Systems and Software* (2016).

Han, Shi, Yingnong Dang, Song Ge, Dongmei Zhang, and Tao Xie. "Performance debugging in the large via mining millions of stack traces". In Proceedings of the 34th International Conference on Software Engineering (ICSE '12). IEEE Press, Piscataway, NJ, USA, , pp. 145-155, 2012.

Highsmith, Jim, and Alistair Cockburn. "Agile software development: The business of innovation." *Computer* 34, no. 9 (2001): 120-127.

Karvonen, Teemu, Lucy Ellen Lwakatare, Tanja Sauvola, Jan Bosch, Helena Holmström Olsson, Pasi Kuvaja, and Markku Oivo. "Hitting the Target: Practices for Moving Toward

Innovation Experiment Systems." In *Software Business*, pp. 117-131. Springer International Publishing, 2015.

Laage-Hellman, Jens, Frida Lind, and Andrea Perna. "Customer involvement in product development: an industrial network perspective." *Journal of Business-to-Business Marketing* 21, no. 4 (2014): 257-276.

North, Dan. "Introducing bdd." *Better Software, March* (2006).

Olsson, Helena Holmström, and Jan Bosch. "The HYPEX Model: From Opinions to Data-Driven Software Development." In *Continuous Software Engineering*, pp. 155-164. Springer International Publishing, 2014.

Olsson, Helena Holmström, and Jan Bosch. "Towards continuous validation of customer value." In *Scientific Workshop Proceedings of the XP2015*, p. 3. ACM, 2015.

Olsson, Helena Holmström, Hiva Alahyari, and Jan Bosch. "Climbing the" Stairway to Heaven" - A Mulitiple-Case Study Exploring Barriers in the Transition from Agile Development towards Continuous Deployment of Software." In *Software Engineering and Advanced Applications (SEAA), 2012 38th EUROMICRO Conference on*, pp. 392-399. IEEE, 2012.

Pachidi, Stella, Marco Spruit, and Inge van de Weerd. "Understanding users' behavior with software operation data mining'. Computers in Human Behavior, Vol 30, pp. 583–594, 2014. Elsevier.

Ries, Eric. "The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. 2011." *Crown Business, NY, USA*.

Rissanen, Olli, and Jürgen Münch. "Continuous experimentation in the B2B domain: a case study." In *Proceedings of the Second International Workshop on Rapid Continuous Software Engineering*, pp. 12-18. IEEE Press, 2015.

Robson, Colin. “Real world research: A resource for users of social research methods in applied settings.” Wiley, Chichester, 2011.

Runeson, Per, and Martin Höst. "Guidelines for conducting and reporting case study research in software engineering." *Empirical software engineering* 14, no. 2 (2009): 131-164.

Srivastava, Jaideep, Robert Cooley, Mukund Deshpande, and Pang-Ning Tan. "Web usage mining: discovery and applications of usage patterns from Web data". ACM SIGKDD Explorations Newsletter, Vol 1, Issue 2, pp. 12-23, 2000.

Steiber, Annika, and Sverker Alänge. “A Corporate System for Continuous Innovation: The case of Google Inc.” European Journal of Innovation Management, pp. 243–264, 2013.

Tang, Diane, Ashish Agarwal, Deirdre O'Brien, and Mike Meyer. "Overlapping Experiment Infrastructure: More, Better, Faster Experimentation". In Proceedings of the 16th Conference on Knowledge Discovery and Data Mining, pp. 17–26, 2010.

Tichy, Matthias, Jan Bosch, Michael Goedicke, and Brian Fitzgerald. "2nd international workshop on rapid continuous software engineering (RCoSE 2015)." In *Proceedings of the 37th International Conference on Software Engineering-Volume 2*, pp. 993-994. IEEE Press, 2015.

Yaman, Sezin Gizem, Fabian Fagerholm, Myriam Munezero, Jürgen Münch, Mika Aaltola, Christina Palmu, and Tomi Männistö. "Transitioning Towards Continuous Experimentation in a Large Software Product and Service Development Organisation–A Case Study.", To appear, *In International Conference on Product-Focused Software Process Improvement (PROFES)*, 2016

Yin, Robert. "Case study research: Design and methods" (4th Ed.). Thousand Oaks, CA: Sage, 2009.

Walsham, Geoff. "Interpretive case studies in IS research: nature and method." *European Journal of information systems* 4, no. 2 (1995): 74-81.