Transitioning Towards Continuous Experimentation in a Large Software Product and Service Development Organisation – A Case Study

Sezin Gizem Yaman[1](✉), Fabian Fagerholm[1], Myriam Munezero[1], Jürgen Münch[1,2], Mika Aaltola[3], Christina Palmu[3], and Tomi Männistö[1]

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

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

[3] Ericsson, Hirsalantie 11, 02420 Jorvas, Finland
{mika.aaltola,christina.palmu}@ericsson.fi

Abstract. *Context:* Companies need capabilities to evaluate the customer value of software-intensive products and services. One way of systematically acquiring data on customer value is running continuous experiments as part of the overall development process. *Objective:* This paper investigates the first steps of transitioning towards continuous experimentation in a large company, including the challenges faced. *Method:* We conduct a single-case study using participant observation, interviews, and qualitative analysis of the collected data. *Results:* Results show that continuous experimentation was well received by the practitioners and practising experimentation helped them to enhance understanding of their product value and user needs. Although the complexities of a large multi-stakeholder business-to-business (B2B) environment presented several challenges such as inaccessible users, it was possible to address impediments and integrate an experiment in an ongoing development project. *Conclusion:* Developing the capability for continuous experimentation in large organisations is a learning process which can be supported by a systematic introduction approach with the guidance of experts. We gained experience by introducing the approach on a small scale in a large organisation, and one of the major steps for future work is to understand how this can be scaled up to the whole development organisation.

Keywords: Continuous experimentation · Experiment-driven software development · Product management · Lean startup · Customer development · Customer involvement · Organisational transition · Agile software development · Case study

P. Abrahamsson et al. (Eds.): PROFES 2016, LNCS 10027, pp. 344–359, 2016.
DOI: 10.1007/978-3-319-49094-6_22

1 Introduction

Continuous experimentation is a software development approach where research and development (R&D) activities are driven by constantly conducting experiments with product value [1–3]. Customers and users are involved in the decision-making process as experiment subjects, providing data by interacting with experiment materials, such as the software features being developed or related design artefacts. Product value is tested by observing actual behaviour rather than relying on secondary sources, opinions, or assumptions.

Although several approaches to experiment-driven software development have been proposed (e.g. [1,2,4]), guidance is lacking on how development teams in large organisations with complex business partnership networks can adopt them. In this paper, we investigate the introduction of continuous experimentation in a large software development organisation in a B2B domain. We observe different roles, means of communication, and integration with the overall development process. Furthermore, we investigate how customers and users are accessed and involved. We collect observed challenges and lessons learned that arise when the teams attempt to perform experiments to support decision-making. More specifically, we seek to answer the following research question:

RQ: How can a large software development organisation transition towards continuous experimentation in a B2B domain?

In order to address the research question, we conducted a single-case study in which we observed and participated in the introduction of continuous experimentation in a large company. Two teams, a development and a UX team, collaborated to select a target for experimentation and to design and implement an experiment to help make a focused product decision. Through the case study, we uncovered some of the critical factors that may support or impede the transition.

The rest of this paper is structured as follows. Section 2 presents the background and related work relevant to this study. Section 3 describes the research approach, including the context in which the case study was conducted, and the data collection and analysis methods. The design and execution details of an experiment conducted by the case company are detailed in Sect. 4. The transition process towards continuous experimentation is outlined in Sect. 5. The findings are discussed and the research question is addressed in Sect. 6. Section 7 concludes the paper and highlights potential future work.

2 Background and Related Work

Considering product value as a first-class concept in software development was proposed in value-based software engineering (VBSE) [5]. VBSE asserts that instead of treating software engineering as value-neutral, its major artefacts and activities should be analysed to assess what value they provide to customers and users, and use knowledge of that value in decision-making. Value has also been considered in agile software development [6,7] and in approaches to product development and entrepreneurship such as Lean Startup [8], Customer Development [9], and Lean Analytics [10]. A body of literature is emerging in software research that addresses this and related topics. In this section, we review a selection from this set of related work.

To survive and compete in today's fast-changing development environments, organisations have to develop, release, and learn from their software products and services quickly [11]. Hence, many software companies have adopted or are adopting agile practices, which champion flexibility, efficiency, and speed in developing software [6]. Nevertheless, Holmström Olsson et al. [12] suggest that the application of agile methods in software R&D activities is only one stage on the maturation path of companies' software engineering practices. At the final stage of the model – R&D as an experiment system – development is based on rapid experiments that utilise instant customer feedback and product usage data to identify customer needs.

The experiment-driven stage of software product and service 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 opinions [1–3,13]. Through experiments, organisations can gain evidence about which features customers actually want, thus helping them to avoid developing features that are not valuable to customers [4]. As Bosch [14] states, "the faster the organisation learns about the customer and the real-world operation of the system, the more value it will provide."

Continuous experimentation may take different forms in different environments. Rissanen and Münch [3] list a number of customer-related challenges that continuous experimentation faces in B2B domains. For instance, customers may have to be informed in advance and sign a written agreement to participate in experiments. End users are not always the customers of the organisation, but they can be a customer's customer. Pro-active lead customers might have to be involved in the experiment design process, but may be challenging to acquire. Also, it may not be possible to interrupt the daily work of users in order to involve them in experiment tasks.

Thus, how to integrate experimentation in the software product development cycle is still a key question. Fagerholm et al. [2] propose the RIGHT model for continuous experimentation. The model consists of a process model and an architectural model. In the process model, assumptions are first identified, experiments are designed to test them, experiment materials (such as minimum viable features) are built, the experiment is executed, and analysis results are then used to support product development decision-making. The decision may be to fully develop and deploy a feature or to pivot if the experiment indicates that the feature is unsuccessful. The architectural model outlines additional infrastructure that is required to carry out such experiments continuously, in parallel, and at scale. In this study, we are guided by the fundamentals of the RIGHT model in the introduction of continuous experimentation.

3 Research Approach

This study follows a holistic single-case study approach [15] in order to gain deeper understanding of how development teams in a large organisation adopt continuous experimentation. Additionally, the study has elements of action research, in that the researchers were actively involved with the process being studied [16]. The unit of analysis is the process of transitioning towards continuous experimentation. We observe only the start of the transition, but consider this unit of analysis to be bounded by an identifiable starting point, and potentially ending in either non-adoption or adoption to different degrees. The transition process may be considered to concern several parts of the organisation, but our observation is limited to one unit concerned with product development. The data collection phase took place over a three-month period in autumn 2015.

3.1 Case Context

The company involved in the case study is a global corporation specialising in providing communication technology and services. The organisation is highly distributed, with globally allocated development teams. This study is conducted in the context of a connectivity management and billing service platform that the company develops for telecom operators and their enterprise customers. This platform includes a management portal, used by operator users, which is the focus of this study.

Figure 1 illustrates the parties and their location in the B2B network, revealing a multi-layer structure of stakeholders. The platform development project involves 11 teams, with around 70 people, who are distributed over multiple locations globally. The unit of observation in this study is one software development team and one UX team located in Finland, who are working on the aforementioned management portal.

The teams are incrementally developing a new version of the portal, which includes modernising the visual design and functionality. While the purpose is to keep the current set of functionality, enhancements to user workflows can be made if this does not impede the delivery schedule.

At the time of the study, the two teams were tasked with implementing an activity log inside the portal which would provide 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. The activity log was chosen for this study both because it was the teams' next assignment, and because there were open questions regarding its design.

3.2 Research Process

The study was conducted in an iterative fashion, with company representatives evaluating decision points, executing the experiment, collecting and analysing

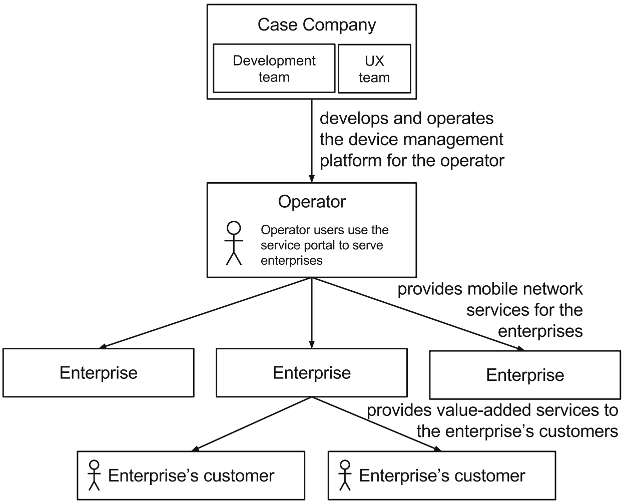

Case Company
Development team
UX team
develops and operates the device management platform for the operator
Operator
Operator users use the service portal to serve enterprises
provides mobile network services for the enterprises
Enterprise
Enterprise
Enterprise
provides value-added services to the enterprise's customers
Enterprise's customer
Enterprise's customer

Fig. 1. The case company and other actors in the B2B network formed around the platform. For clarity, only one operator is shown, although there are multiple operators.

the experiment data, and with researchers observing the process, analysing the collected research data, and proposing alternative decision paths. An initial meeting was held where the principles of continuous experimentation and the RIGHT model [1,2] were explained to development and UX teams and product owners. After reaching a positive decision from the company, the joint collaboration proceeded. Multiple meetings were held, both online and face to face, to (1) understand the case context, (2) explore and select an experiment target, (3) identify assumptions related to that target, (4) develop a hypothesis and experiment design, (5) discuss operational details regarding experiment execution, (6) analyse experiment data, (7) draw conclusions based on the analysis and (8) plan the next steps. Between meetings, materials from previous iterations were analysed and developed to support subsequent decisions and actions.

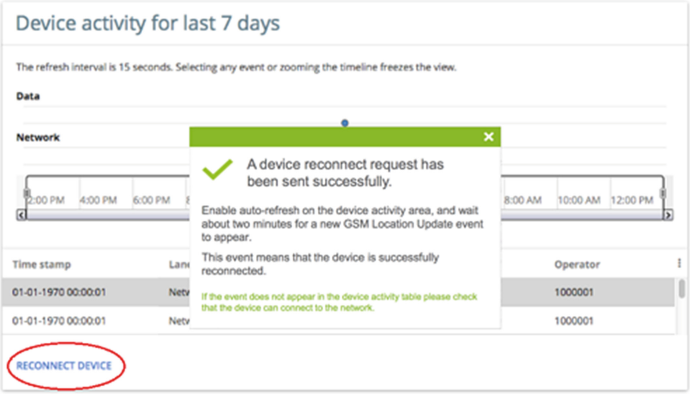
This study uses materials produced in and for the meetings as well as other primary data sources, which include participant observation, transcripts of audio recordings of face-to-face meetings, minutes and notes of meetings (both at the customer site and online, including weekly online status meetings), open-ended semi-structured interviews, email communication and background material from the company. In total, there were three on-site and eight remote meetings. The accumulated material was analysed using thematic analysis [16,17]. The data was first extracted and analysed to form initial themes. These were then cross-checked against the gathered materials and refined into final themes which are presented and discussed in Sect. 5.

4 Designing and Executing the Experiment

As our aim was to observe the introduction of continuous experimentation in a company, we conducted an actual experiment round with a real product, i.e., the activity log described in Sect. 3.1. Here, we did not seek to reach a valid and generalisable result in the scientific sense, but rather to obtain enough evidence to support a technical decision. In this section, we describe the process of designing and executing the experiment.

The experiment was planned by a technical coach from the development team, two people from the UX team, and three researchers. The first decision to be made was to select a target for the experiment by analysing the feature requirements for the activity log. Behaviour-driven development (BDD) stories [18] were developed and analysed during the study in order to better understand the user requirements associated with the activity log.

In total, seven BDD stories pertaining to the activity log were analysed. With each BDD story, underlying assumptions regarding user needs and behaviour were identified. From the identified assumptions, hypotheses to be tested were formed. Subsequently, proposals of experimental designs to validate the hypotheses were drafted. From these, the development and UX teams selected one design proposal to be the experiment target, which was then elaborated into a more complete experiment design.



Device activity for last 7 days

The refresh interval is 15 seconds. Selecting any event or zooming the timeline freezes the view.

A device reconnect request has been sent successfully.

Enable auto-refresh on the device activity area, and wait about two minutes for a new GSM Location Update event to appear.

This event means that the device is successfully reconnected.

If the event does not appear in the device activity table please check that the device can connect to the network.

Time stamp	Lan	Operator
01-01-1970 00:00:01	Netv	1000001
01-01-1970 00:00:01	Netv	1000001

Fig. 2. Mockup of activity log with “reconnect” button indicated and feedback message displayed.

The selected experiment tested options for a feedback message that is displayed after operator users click on a “reconnect” button in the activity log (see Fig. 2). The reconnect button sends a request to the mobile network, asking it to flush the current SIM card registration, which means the mobile device must reconnect in order to resume normal operation. This action can be used to recover from certain error conditions. As the mobile network provides no feedback on the request, the reconnect status cannot be accurately displayed to users in the activity log. This might lead to a situation where a user clicks the button several times to no avail. Thus a good feedback message would inform a user on the current state of the system as well as what to do next, while a bad feedback message would result in increased load on the network, delays in problem resolution, worse experience for all users involved, and potential costs associated with these negative effects.

A series of user interface mockups with feedback messages were created for the experiment. These were first piloted with the product owner and updated based on the feedback given. After the update, the experiment was run with test subjects. Two runs of the experiment were conducted as illustrated in Table 1. In the first run, the experimenters realised that there were flaws in the mockups and feedback messages – they were unclear and misleading to the test subjects. Additionally, the experimenters had difficulties determining whether a user succeeded according to the criteria outlined in the hypothesis (see Table 1). Hence, the mockups and feedback messages were updated and the experiment was rerun. In the second run, the original reconnect feedback message was also included. In both of the runs, the order of the message candidates was balanced so that each message appeared at least once and the order changed for each test subject in order to avoid the risk of a learning effect biasing the results.

From the data analysis of the second run, one feedback message (message 6 in Table 2) had the highest score, on criterion 1, which was prioritised by the teams. It also scored well on the other criteria. Message 6 was thus selected for inclusion in the next product release. The results also revealed that the original message (message 7) performed poorest on all three criteria.

5 Transitioning Towards Continuous Experimentation

In the process of planning, designing, executing, and analysing the experiment, a number of observations and inferences were made regarding the transition towards continuous experimentation both from practitioners' and researchers' points of view. In this section, we present these findings under the themes that were deduced from our data analysis.

5.1 Initial Circumstances

Prior to the decision to proceed with experiment-driven software development, we observed an initial interest towards continuous experimentation among company representatives, but also concerns as the adoption process started in the middle of development with an evolving product. The product owner wanted to limit risks while practising with the new approach. This raised some important questions: is it possible to start at the team level and with small-scale experiments in order to gain experience before scaling up to multiple teams, higher in the organisation, and experiment targets that have a larger impact on the

Table 1. Experiment details for the first and second experiment run.

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	
	First run	**Second run**
Hypothesis	We believe that with the right feedback message, users should be able to tell: (1) what the state of the device connection is and (2) what the next action is. In order to validate this, users will be shown a set of feedback messages and will be asked to provide answers to the above two criteria. The message with the most "yes" answers for each criterion will be the best message and will be selected	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 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
Minimum viable feature	Five mockups (PowerPoint) with different feedback messages	Seven mockups (PowerPoint) with different feedback messages
Test subjects	Three internal company employees invited by the experimenters based on availability	Seven internal company employees invited by the experimenters based on availability
Experimenters	One person from development team and one from UX team	One person from development team, one from UX team and an additional observer from the UX team (present only in some sessions)
Collected data	Yes or no scores for each test subject according to each hypothesis criterion, experimenters' observations of test subjects during the experimentation, and unstructured interview notes	
Duration (total)	60 min	120 min
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	

system being developed? Moreover, existing release deadlines dictated the target and scale of the experiment, as well as the resources that could be allocated to it.

Table 2. Scores for each feedback message with the winning message highlighted. Each test subject was exposed to seven message candidates and was scored by two experimenters. Criterion 1 had double the weight when choosing the winner. (Note: There was a data entry error for message 5 where one test subject's scores on criterion 2 and 3 were not recorded. However, this does not impact the result of the experiment.)

Feedback message candidate	**Criterion 1** (weight: 2×)		**Criterion 2** (weight: 1×)		**Criterion 3** (weight: 1×)	
	Yes	No	Yes	No	Yes	No
1	7	7	7	7	4	10
2	9	5	10	4	3	11
3	5	9	8	6	2	12
4	5	9	10	4	2	12
5	7	7	8	4	6	6
6	13	1	11	3	4	10
7	4	10	5	9	3	11

5.2 Starting with Small Teams

We observed that beginning with small teams who are interested did facilitate the introduction of a new way of working in the large case organisation. The development team consisted of four developers and the UX team consisted of two persons. Each team had an active person, a "champion", who took the lead in conducting the experiment and communicating the approach to other team members.

While it was possible to get a quick, low-risk start by beginning with small, motivated teams, we observed challenges which might impact scaling of continuous experimentation. For instance, we observed that organisational factors influenced the ease at which experimentation targets could be identified. The necessary product requirements were not always available at the team level. We furthermore observed that limitations in the teams' area of influence affected the experimentation activities. For instance, the decision to involve real users in conducting the experimentation required approval from different management levels and extra consideration since most of the customers were abroad. Also, dependencies on other teams and release management decisions meant that product changes based on the experiment result could not be immediately integrated into the next release, but into the succeeding one.

5.3 Small-Scale Experiments

From the time that BDD stories were developed to the analysis of the results, while the experiment planning process took approximately one month, executing the experiment only took a couple of hours (see Table 1). Nevertheless, the aim was to initiate the experimentation activity and to learn how to experiment, i.e., to "experiment with experimentation" as the technical coach put it. He also added that “[It’s better to] start experimenting with something small. [...] It’s more important to start now. Practice will make it perfect.”

As continuous experimentation is a way to achieve customer and user involvement, user access was a discussion point during the experiment planning stage. The decision to use internal test subjects was mostly a question of time and effort as noted by the UX designer: “It would be really time consuming to contact our actual customers, write emails and explain what this [experiment] is about. [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 for the beginning”. The technical coach also added that “there is a limit to how much you can e.g., interview the customers before you provide something [concrete]”. The team members were aware of the drawbacks of using internal test subjects, but deemed it more important to get started with the first cycle than spending time on accessing users. “Of course we thought of how much [more benefit could be gained by experimenting] with actual customers. But then, this experimentation is about the UX part, [...] and we did not see that we would get much more benefit if we had waited weeks to get real customer input.” (Technical coach).

5.4 Identifying an Experimentation Target

We observed that it was not straightforward to identify an experimentation target. In particular, options tended to be more technical than value-based. During planning, it emerged that there was no clear understanding on some of the platform features, and user requirements were not directly available in written form. Instead, we deduced them from other materials, such as user journeys and personas obtained from user research, and mockups from prototypes, all developed in the beginning of the project. We had numerous discussions with the teams to clarify the purpose of the activity log and its different functions. Finally, BDD stories were developed and utilised to identify assumptions behind the user requirements, and the experiment was derived from those assumptions.

5.5 Designing and Executing the Experiment

The experiment was run with internal company employees. Even though the case company had done a pilot study with a product owner to revise the experiment design, they easily recognised during the first experiment run with test subjects that additional planning was essential. Test feedback messages were unclear and scoring criteria specified in the hypothesis were not explicit enough for experimenters to reach an agreement. Therefore, better background information and clearer instructions for the subjects were developed before running a second round of the experiment. The hypothesis was also revised and clearer tasks for the experimenters were defined. Although some effort should be spent on improving the design and execution, we found that it had to be balanced with available resources. Small-scale experiments especially meant that effort should not be expended beyond what is required to get a sufficient result: “over-planning [improving the experiment beyond a certain point] would be pointless”, according to the technical coach.

5.6 Collaborating with Experts

Expertise was provided by the five researchers involved in the introduction of the continuous experimentation approach. Support for the transition was particularly provided during the planning and execution of the experiments.

At the beginning of the study, the teams and other company representatives had to spend time introducing the product and its context to the experts. However, they stated that it was beneficial to have expert facilitators guiding the transition and providing support and guidance when they needed it. In this case, some mistakes were avoided through expert opinion. For instance, during the execution stage, guidance was provided on how to achieve more valid results and avoid introducing bias during the experiment – e.g. avoiding leading the users by keeping discussions between experimenters and test subjects minimal, and ensuring that there were at least two experimenters.

5.7 Persistence

Continuous experimentation may be easy to understand in principle, but actually starting it in a real, large B2B organisation required persistence. The final experiment design was reached after a number of attempts. The pilot run and two rounds of actual experiment runs were required to obtain data for the final analysis. The teams indicated that when starting, one should not dwell on temporary failures. Better to “fix the experiment [the] best way you can and run it again. You can learn so much with each experiment.” (Technical coach)

Moreover, the teams were willing to include experimentation in some of their standard procedures. They decided to build a wiki library where all experimentation details and learnings would be stored so that the information can be reused when necessary and help guide other teams who want to practice the approach. Also, the champions in the teams were persistent in documenting each step of the process, which helped communication internally and with experts. Thus, some of the prerequisites of scaling the approach to cover a larger portion of the organisation are in place.

Table 3 summarises the challenges faced when transitioning towards experimentation together with observed mitigation strategies under each of the six themes presented in Sects. 5.1, 5.2, 5.3, 5.4, 5.5, 5.6 and 5.7.

6 Discussion

Transitioning towards continuous experimentation is a learning process, at the core of which is the development of the organisational capability to identify

Table 3. Identified challenges and mitigation strategies.

Theme	Challenges	Mitigation strategies
Initial circumstances	– Evolving product, existing plans, deadlines – Limited resources – Need to limit risks	– Allocate only few resources to begin with – Choose a small scope for the initial experiment
Starting with small teams and small-scale experiments	– Higher level product information might not be visible at the team level – The team's area of influence may be limited – Inaccessibility of real users – Experimentation activities may not be initiated because of prior commitments	– Involve people from different teams in brainstorming and planning the experiment together – Utilise resources that are more accessible, e.g. internal company employees – Good to have champions in teams pioneering the transition
Identifying an experimentation target	– Difficult to select the features to start experimenting with – Identified experiment targets may be on a mostly technical level	– Utilising existing product-related materials helps identify experiments, e.g., BDD stories – Having discussions with team members and experts – Carefully analyse the feature to be experimented on to identify user needs and assumptions
Designing and executing the experiment	– The lack of experimenter experience can lead to biases being introduced in the design and execution of experiments – Effort and time for planning needs to be allocated for running a valid experiment	– Piloting and rerunning the experiment helps to enhance the experiment design, and reach more valid results – Seek expert advice to avoid potential biases in the experiment – Overplanning should be avoided; when starting, the important thing is to learn
Collaborating with experts	– Effort is needed for introducing the product and the context to the experts	– Experts can help avoid mistakes in experiment design and execution; the effort to introduce may pay off – Introduction may be sped up by using materials that are already needed in development, such as user stories and requirements expressed as, e.g., BDD stories
Persistence	– First experimentations can be seen as effortful and non-efficient	– Keep practising, learning will increase efficiency – Experiment designs and guidelines for executing experiments can be gathered and reused, reducing future effort

assumptions, test them in experiments, and support development decisions based on the evidence. However, many other factors play a role in the transition both on the team and organisational levels. In this section, we address the research question and compare the study findings with related work.

> *How can a large software development organisation transition towards continuous experimentation in a B2B domain?*

Based on the findings from the case study data analysis, we identified circumstances and activities that can be taken to enable the transition towards continuous experimentation. Even though there might be initial circumstances that constrain the transition, we observed that initiating the transition is possible by starting with small teams and small-scale experiments. In order to lower the barriers to starting, experiment targets can be identified from existing materials. Collaboration with experts can be used for guidance and support but team effort is still needed in planning, designing and executing experiments. At this stage, learning about experimentation is the most important thing. Later, it is essential to find ways to sustain the process, making it continuous, and scaling it to cover a larger part of the development organisation.

6.1 Challenges and Lessons Learned

The biggest benefit gained by the teams was that they learned to perform experimentation in a more systematic way, which will help them to better understand what their customers want and take the right steps in increasing user satisfaction and reduce support costs. Doing experimentation also helped the teams gain new insights and better understanding of their ongoing work. In addition, based on the teams' experiences, they realised that "experimentation made it clear to the team that there is no need to debate between opinions and assumptions as you can quickly test them with an experiment." (Technical coach)

Information about user needs, e.g. requirements, is needed to identify assumptions. In this case, BDD stories proved to be useful for this purpose, but other forms may be possible, as well. The experimentation activity needs to be integrated with the overall development process. Once integrated, the effort put in planning experiments could diminish. In this case, planning the study and the experiment took around a month each, since they included establishing the collaboration between the experts and the teams from the organisation, getting to know the context, and identifying assumptions in the product. On the other hand, running the experiment itself took only hours.

Several challenges arose because of the complex B2B environment. For instance, the path from the development organisation to users through the B2B network was long and involved many organisations. For this reason, end users could not be included in experiments with reasonable effort, a finding that is in line with Rissanen and Münch's [3] observations. Other approaches to get experiment data were needed in order to compensate this challenge. In this case, internal test subjects were an economical way to start developing the capability for continuous experimentation, and made it possible to even gain some decision support for a small product decision although the set of subjects was limited.

Some barriers faced were a result of timing. The process started in the middle of development with an evolving product. Existing release deadlines influenced the resources available, as well as the target and scale of the experiment. The selected product feature to be experimented on was quite small and the experiment was more about optimisation rather than validating a complete feature. However, when the purpose is to practise experimentation, this is not a critical issue, but it must be understood that experiment results might not be the most beneficial or target the most value-creating features in the beginning.

In general, it was difficult to design an experiment that would actually test the value of a feature. This would have entailed determining whether the feature is necessary or suitable for accomplishing a given task that has already been found to fulfil a user need. There are multiple possible reasons for why designing a value-related experiment was difficult, and the long chain from development to user was one of them. It was unclear what the value of different features was and for whom. Also, the long chain meant more uncertainty about how a feature contributes to value. A feature may fulfil a need indirectly, and mapping the chain was not possible in this study.

Another challenge was that it was difficult to design a behavioural experiment task, meaning a task that would test whether a feature contributes to a behaviour change. This would have been necessary in order to determine whether the feature contributes to a user need, since it is through behaviour that the need is fulfilled. In this study, the experiment relied on subjects telling what they would do rather than observing whether they carried out certain actions or not. Part of the reason was the effort and cost of setting up required experiment materials. Observing behaviour requires interactive materials that allow the user to express the behaviour to be observed. Lack of such materials may be a barrier when initiating the transition towards continuous experimentation.

Furthermore, it was observed that the scale of experiments can be adjusted operationally so that little or no development effort is required, for example by using PowerPoint mockups as in this case. On the other hand, conducting small-scale experiments is part of a tradeoff between smoothing the path towards continuous experimentation and reaching the level where experiments directly target customer value.

6.2 Threats to Validity

Researchers' expectancy bias might be a threat in this study. Researchers might interpret the collected data in such a way that it fulfils their expectations. In order to mitigate this threat, participants from the case company were involved in the study data analysis stage and participant validation was used in order to verify the study results.

Researcher triangulation was used to address construct and external validity in terms of accuracy checking. Two researchers first conducted the initial study data analysis, then reviewed the analysis process with a third researcher and later, all five researchers reviewed the results along with discussion sessions. This ensured that the study results would not rely on interpretations of single researchers only.

In terms of generalisability, we are interested in whether the results of this study would be applicable to other software development organisations transitioning towards continuous experimentation. The characteristics of the case company, such as its size, structure, customers, business domain, the scale of the experiment, the product, and other contextual factors may limit the transferability of the results presented in this paper. It is not yet clear how such transfers can be made. Due to the novelty of the field and early maturity of experimentation in the company, there is not much evidence available to support transferring the results. We hope our findings will contribute to the knowledge about transition to continuous experimentation when combined with further research.

7 Conclusion

We conducted a holistic single-case study in a large, global telecom company operating in a B2B environment. We introduced an approach to continuous experimentation in the case company. Two company teams and five researchers conducted a single experiment cycle with internal test subjects. The experiment results allowed the company to make a product development decision which improved the usability of a part of their product. By participating in the activity, we observed the first steps of a transition towards continuous experimentation.

We found that the approach was easy for practitioners to understand and reception was favourable in general. The experiment activity highlighted important questions about the product under development and how it could best serve users. The collaboration between the UX and development teams was also enhanced, as expertise from both was required to plan and execute the experiment. Starting with small teams and experiments with a tightly limited scope allowed a fast start and a short, one-month cycle time from design to results.

We also found several challenges that may hinder the adoption of an experiment-based approach and limit its benefits in the initiating phase. Our study shows that it may be difficult to find an experiment target due to information about user needs and goals being scattered in a large B2B organisation. This makes it difficult to identify the assumptions that should be tested in experiments. It may also be difficult to reach the level where experiments directly address product value rather than optimising usability. Involving users directly in experiments was difficult in this B2B case, and may come with additional cost, but would also make experiments more valid and relevant. Designing experiments around assumptions about the user behaviours that are related to value creation should result in experiments with more impact. This remains a difficult challenge which warrants further research. More research is also needed on how to integrate continuous experimentation with the overall organisation and how this affects culture, architectures, methods, processes, management, and staffing in contemporary organisations.

References

1. Fagerholm, F., Guinea, A.S., Mäenpää, H., Münch, J.: Building blocks for continuous experimentation. In: Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering, RCoSE 2014, pp. 26–35. ACM, New York (2014)
2. Fagerholm, F., Guinea, A.S., Mäenpää, H., Münch, J.: The RIGHT model for Continuous Experimentation. J. Syst. Softw. (2016, in press). doi:10.1016/j.jss.2016.03.034.
3. Rissanen, O., Münch, J.: Continuous experimentation in the B2B domain: a case study. In: Proceedings of the Second International Workshop on Rapid Continuous Software Engineering, RCoSE 2015, Piscataway, NJ, USA, pp. 12–18. IEEE Press (2015)
4. Holmström Olsson, H., Bosch, J.: The HYPEX model: from opinions to data-driven software development. In: Bosch, J. (ed.) Continuous Software Engineering, pp. 155–164. Springer, Cham (2014)
5. Boehm, B., Huang, L.G.: Value-based software engineering: a case study. Computer **36**(3), 33–41 (2003)
6. Highsmith, J., Cockburn, A.: Agile software development: the business of innovation. Computer **34**(9), 120–127 (2001)
7. Cockburn, A., Highsmith, J.: Agile software development, the people factor. Computer **34**(11), 131–133 (2001)
8. Ries, E.: The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Business, Houston (2011)
9. Blank, S.: The Four Steps to the Epiphany: Successful Strategies for Products That Win, 2nd edn. K&S Ranch, Pescadero (2013)
10. Croll, A., Yoskowitz, B.: Lean Analytics: Use Data to Build a Better Startup Faster. O'Reilly Media, Sebastopol (2013)
11. Tichy, M., Bosch, J., Goedicke, M., Fitzgerald, B.: 2nd International Workshop on Rapid Continuous Software Engineering (RCoSE 2015). In: Proceedings of the 37th International Conference on Software Engineering, vol. 2, pp. 993–994. IEEE Press (2015)
12. Holmström Olsson, H., Alahyari, H., Bosch, J.: Climbing the "Stairway to Heaven" - a mulitiple-case study exploring barriers in the transition from agile development towards continuous deployment of software. In: 2012 38th Euromicro Conference on Software Engineering and Advanced Applications, pp. 392–399 (2012)
13. Yaman, S.G., Sauvola, T., Riungu-Kalliosaari, L., Hokkanen, L., Kuvaja, P., Oivo, M., Männistö, T.: Customer involvement in continuous deployment: a systematic literature review. In: Daneva, M., Pastor, O. (eds.) REFSQ 2016. LNCS, vol. 9619, pp. 249–265. Springer, Heidelberg (2016). doi:10.1007/978-3-319-30282-9_18
14. Bosch, J.: Building products as innovation experiment systems. In: Cusumano, M.A., Iyer, B., Venkatraman, N. (eds.) ICSOB 2012. LNBIP, vol. 114, pp. 27–39. Springer, Heidelberg (2012). doi:10.1007/978-3-642-30746-1_3
15. Yin, R.: Case Study Research: Design and Methods, 4th edn. SAGE Publications, Inc., Thousand Oaks (2009)
16. Robson, C.: Real World Research. Wiley, Chichester (2011)
17. Braun, V., Clarke, V.: Using thematic analysis in psychology. Qual. Res. Psychol. **3**(2), 77–101 (2006)
18. North, D.: Introducing BDD. Better Software., March 2006