2015 IEEE/ACM 2nd International Workshop on Rapid Continuous Software Engineering

Continuous Experimentation in the B2B Domain: A Case Study

Olli Rissanen*†, Jürgen Münch†
*Steeri Oy, Tammasaarenkatu 5, 00100, Helsinki
†Department of Computer Science, University of Helsinki
P.O. Box 68, FI-00014 University of Helsinki, Finland
Email: olli.rissanen@aalto.fi, juergen.muench@cs.helsinki.fi

***Abstract*—Rapid value delivery requires a company to utilize empirical evaluation of new features and products in order to avoid unnecessary product risks. This helps to make data-driven decisions and to ensure that the development is focused on features that provide real value for customers. Short feedback loops are a prerequisite as they allow for fast learning and reduced reaction times. Continuous experimentation is a development practice where the entire R&D process is guided by constantly conducting experiments and collecting feedback. Although principles of continuous experimentation have been successfully applied in domains such as game software or SAAS, it is not obvious how to transfer continuous experimentation to the business-to-business domain. In this article, a case study from a medium-sized software company in the B2B domain is presented. The study objective is to analyze the challenges, benefits and organizational aspects of continuous experimentation in the B2B domain. The results suggest that technical challenges are only one part of the challenges a company encounters in this transition. The company also has to address challenges related to the customer and organizational culture. Unique properties in each customers business play a major role and need to be considered when designing experiments. Additionally, the speed by which experiments can be conducted is relative to the speed by which production deployments can be made. Finally, the article shows how the study results can be used to modify the development in the case company in a way that more feedback and data is used instead of opinions.**

I. Introduction

To deliver value fast and to cope with the increasingly active business environment, companies have to find solutions that improve efficiency and speed. Agile practices have increased the ability of software companies to cope with changing customer requirements and changing market needs [9]. To even further increase the efficiency, shortening the feedback cycle enables faster customer feedback. Additionally, providing software functionality to collect usage data can be used to guide the product development and to increase the value generated for customers. Eventually the company running the most and fastest experiments might outcompete others by having the data to develop products with exceptional qualities [6].

The requirement of developing value fast has led to the evolution of a new software development model [3]. The model is based on three principles: evolving the software by frequently deploying new versions, using customers and customer data throughout the development process and testing new ideas with the customers to drive the development process and increase customer satisfaction. Currently more and more software companies are transitioning towards this model, called innovation experiment systems. This development model shares the key principles with continuous experimentation.

Continuous experimentation is a term created by Dan McKinley, a Principal Engineer at Etsy [4]. Conducting continuous experiments addresses the problem of organizations having many ideas, but the return-on-investment might be unclear and the evaluation itself might be expensive [11]. Applying continuous experimentation reduces the amount of guesswork and instead provides a systematic approach towards product development. Kaushik states in his experimentation and testing primer that *"80% of the time you/we are wrong about what a customer wants"* [14]. The company Netflix has also found similar results, considering 90% of what they try to be wrong [15].

This study presents a case study exploring how continuous experimentation can be applied in the case company. While existing studies of applying the development model exist [2], [3], [12], [5], only a single study focuses on the B2B domain [19]. This study specifically aims to identify the main requirements, problems and key success factors with regards to this approach in the B2B domain.

This study is organized as follows. The second section summarizes the relevant literature and theories to position the research and to educate the reader on the body of knowledge and where the contributions are intended. The third section presents the research design. The findings are then presented in the fourth section, organized according to the research questions. Finally, the fifth section summarizes the results of study, discusses the limitations of the study and introduces further research avenues.

II. Related work

Fagerholm et al. describe the basic building blocks of continuous experimentation [12]. The authors define that a suitable experimentation system must be able to release minimum viable feature (MVF) or products (MVP) with suitable instrumentation for data collection, design and manage experiment plans, link experiment results with a product roadmap and manage a flexible business strategy. The development cycle in continuous experimentation resembles the Build-Measure-Learn cycle of Lean Startup [13].

978-1-4673-7067-7/15 $31.00 © 2015 IEEE
DOI 10.1109/RCoSE.2015.10

Olsson et al. sketch the typical evolution path of companies [1]. The final stage of the evolution phase is when development is driven by the real-time customer usage of software. Defined as *"R&D as an 'experiment system'"*, the entire R&D system acts based on real-time customer feedback and development is seen as addressing to customers present needs. At the same time deployment of software is seen as a *"starting point for 'tuning' of functionality rather than delivery of the final product"*. While the evolution from continuous delivery to innovation system wasn't explored, Olsson et al. anticipate that the key actions required are the automatic data collection components and capability to effectively use the collected data.

Bosch has widely studied innovation experiment systems as a basis for development [3]. According to Bosch, *"The faster the organization learns about the customer and the real world operation of the system, the more value it will provide"*. The process in innovation experiment systems is to first form a hypothesis that is typically based on business goals and customer "pains" [3]. After the hypothesis has been formed, quantitative metrics to measure the hypothesis must be decided. After this, a minimum viable feature or minimum viable product can be developed and deployed. The software is then run for a period of time while collecting the required data. Finally, the data is analyzed to validate the hypothesis, and a decision is made to either persevere or pivot the business strategy. In simple terms, to persevere with a chosen strategy means that the experiment proved the hypothesis correct, and the full product or feature can be implemented. However, if the experiment proved the hypothesis wrong, the strategy is changed based on the implications of a false hypothesis.

III. CASE STUDY DESIGN

The presented study is designed as a single-case study with the development process as unit of analysis. The reasons for a single-case design are that the research questions mainly address the process and that the relevance of specific context characteristics (such as team characteristics) are not yet sufficiently understood to set up a multi-case design. Two teams from the case company are studied that develop two different software products. The first product, a marketing automation called Dialog, is used through an extensive user interface. The second product under inspection is CDM, a master data management [10] solution running as an integrated background web application. The unit of analysis is studied by focusing on interviewing individual members at different positions in the teams. The objective of the study can be summarized as follows:

Research objective Analyze the software development process transition towards a shorter feedback cycle and data-driven decisions with respect to challenges, benefits and organizational aspects of continuous experimentation from the perspective of the case company in the context of the B2B domain.

RQ1: What are the B2B specific challenges of Continuous Experimentation?
Software development practices and product characteristics vary with the domain and delivery model. Typical business-to-customer (B2C) applications are hosted as Software as a Service (SaaS) applications, and accessed by users via a web browser. In the B2B domain, applications installed to customer environments are very common. Running experiments on customer environments usually requires a new deployment of the software product, possibly causing downtime and changing some functionality of the software. This research question aims to identify the challenges faced in applying this practice in the B2B environment. The research question is answered by conducting interviews to discover the current development process and its challenges in the case company, and using these findings and the available literature on continuous experimentation to map the initial set of challenges this approach will face in the case company. The available literature is used to provide a thorough understanding of continuous experimentation as a whole, so that challenges can be identified in all aspects of the practice.

RQ2: How does Continuous Experimentation benefit the case company?
To rationalize the decision to adopt continuous experimentation in a company, the actual benefits to the business have to be identified. This research question aims to identify clear objectives for what can be achieved by adopting this practice. Sections of the interview aim to identify the current perceived problems of the case company related to product development, collecting feedback and guiding the development process. These problems are then compared to the benefits of this approach found from literature.

RQ3: How can Continuous Experimentation be systematically introduced in the B2B domain?
This research question investigates organisational aspects of applying continuous experimentation in the context of the case company. This question is answered by analyzing a continuous experimentation model by Fagerholm et al. [12] in the context of the case company to identify possible deviations caused by the domain and software products.

The primary source of information in this research are interviews performed within the two teams under study. The interviews consists of pre-defined themes focusing on current development process, current deployment process, current interaction with customers, properties of the software products and future ways with continuous experimentation. The purpose of the interviews is to identify the pain points and ways of working in the current development process. This information is then used to draft the initial set of requirements and benefits for continuous experimentation. Data is also collected through the product description documents and development process documents to verify and supplement the interview data. Data triangulation is implemented by interviewing multiple individuals in different positions of the teams. Methodological triangulation is implemented by collecting documentary data regarding the development process from internal documents, and by including observations made by the author in the analysis.

Each interview is a semi-structured interview with a standardized set of open-ended questions, which allows deep exploration of studied objects [8]. The interviews are performed once with every interviewee. There are a total of 12 interviewees: 6 in each team. The interviewees in the first team consist of 5 software designers and one team leader. In the second team, the interviewees consist of 3 software designers, a quality assurance engineer, a manager for commercialization and a team leader.

The data analysis is based on template analysis, which is a way of thematically analysing qualitative data [7]. The initial template was first formed by exploring the qualitative data in two themes: development process and deployment of software. Through multiple iterations of the data, multiple subthemes were then added to the two existing themes by further coding the data.

IV. Results

This section presents the identified challenges of continuous experimentation in the B2B domain according to RQ1. The challenges are structured along technical, organizational, and customer-oriented aspects. Afterwards, the expected benefits of continuous experimentation in the B2B domain are described according to RQ2. Finally, a proposal for systematically introducing continuous experimentation in the case company is described according to RQ3.

A. Technical challenges

The technical challenges are analyzed based on both the product description documents and sections of the interviews focusing on experimentation and feedback. These are then used as a basis for analysing challenges influencing continuous experimentation.

TABLE I
Technical challenges in continuous experimentation.

Specific problem
Not having a user interface limits the scope of experiments
Low end-user volume limits experiments that rely on statistical significance
Product that is only usable via API calls has a limited scope of experiments
Measurable metrics of the software products that provide value for the customer are not extensively identified
Experimentation infrastructure has to be implemented on top of a mature project
Data has to be collected and stored in customer environments, and transferred from the environment for analysis

The case company's products represent two different styles of software products. While Dialog is used by users via a user interface, CDM is used by other systems via a REST API. As CDM is a software component which is primarily used by other systems via APIs and certain calls, tracking user behavior does not provide as much input as with a user interface. While it is possible to experiment with a software component that has a UI and human users by employing A/B testing, other experiments have to be conducted on software without direct user access. Such experiments include feature alpha, usage metrics, usage behavior tracking and participatory design, among others [3].

The user amount also affects the types of suitable experiments. Dialog has on average five users per customer, while CDM has no human users at all. Experiments relying on statistical significance, such as A/B testing and multi-variate testing, lose most of their value with a low user volume. To be able to confirm validations with such experiments requires a varying amount of users depending on the conversion rate of the experiment. The company then either has to work with a lower confidence interval, or have a very high conversion rate to reduce the amount of subjects needed for statistical importance.

The business model for the products affects the target metrics. Since the purpose of the software products is not to generate revenue per se, knowledge of how the customers benefit from the product is required before the metrics can be chosen. The interviewees were only able to name a few metrics that they perceived as target metrics. In B2B product development, the software development is especially focused on adding value for the customer, but eventually the features must also add value for the software product itself.

It is common in B2C applications to fluctuate between different Overall Evaluation Criterias (OEC), conducting multiple experiments with different metrics [11]. In B2B domain the possibility to co-operate with the customer in order to identify the important metrics should be considered, especially since many companies measure internal metrics. Without being able to articulate what the purpose of the experiment is, it is impossible by the organization to determine whether the experiment is succesful [16]. Additionally, a common pitfall in conducting experiments is to pick an OEC *"for which it is easy to beat the control by doing something clearly wrong from a business perspective"* [17].

The case company's software products have been developed for a relatively long time, and the codebase is becoming large and mature. To be able to collect customer or performance data, and to determine customer preference and interest, architectural changes of the product are often required. One of the most important components that has to be added is the instrumentation layer, which is used to collect the data of the experiments, and it can be implemented in multiple ways. Architectural changes are more costly for matured software, which already have a structure and many architectural dependencies.

The case company's products are mostly installed to customer environments, with customer-owned servers. The data has to be effectively stored and acquired from an external environment, if the analysis is to be done at the company's own servers. Accessing and storing the data in the customer environment might cause problems with space, configurations and performance. However, the case company's applications deal with millions of records of data, and the developers don't see storing additional experimentation-related data as a challenge.

B. Organizational challenges

The organizational challenges are analyzed based on the data collected from the experimentation section of the interviews.

TABLE II
ORGANIZATIONAL CHALLENGES IN CONTINUOUS EXPERIMENTATION

Specific problem
Requirements by customers bring in revenue, and are prioritized over own product development
Competence in experimentation is not high, and requires education
Transitioning towards experimentation culture affects other teams, including sales
Creating a culture of innovation within the organization

Transitioning towards an experimental R&D is perceived as desirable but expensive. Product development has been in recession with the accelerating customer amount and increasing customer requirements. The teams under inspection also have no prior experience with experimenting or data analysis, and require training in order to gain domain knowledge of experimentation.

The interviewees are also concerned about organizational divergence, especially if continuous experimentation is only adapted in two teams. Continuous experimentation affects the customer interaction process, and other teams interacting with the case teams need to understand the experimentation concept.

The teams under study have been working with agile and lean practices for a long time, with minimal focus on innovation. Adopting continuous experimentation shifts the mindset more towards innovation and emphasizes the importance of deeply learning the customers needs. Systematically applying experiments also changes the way software products are developed. New features and product ideas are developed with keeping the possibility of confirming validations with experimentations in mind from a very early stage. This results in more measurement, tuning small details and especially making decisions based on data. The interviewees expressed an interest to shift other parts of the organization towards innovational mindset, especially the product management and sales teams, since they are working in the interface to the customer.

C. Customer challenges

The customer challenges are analyzed based on three sections of the interviews: customer interaction, feedback collection and experimentation.

In the case company, features have been primarily created for actual customer requirements. This somewhat limits the direction of product development. In the case company, recently almost 100% of the completed features have been determined

TABLE III
CUSTOMER CHALLENGES IN CONTINUOUS EXPERIMENTATION.

Specific problem
Large part of developed features are requirements from customers
Customers perform routine tasks that cannot be interrupted
Customers have to be informed of major changes
The software has to be acceptance tested by the customers before production release
The end-users might be customers' customers, complicating feedback collection
Proper legal agreements need to be made with the customers to collect usage data and user behavior
A pro-active lead customer is required to develop the experimentation process

by the customer. The product roadmap has advanced by being led by the customer projects, since the revenue comes from the customers. As the product development is in the ideal case guided by both the customer needs and own ideas, the proportion of data-driven decisions is smaller than in B2C.

Another issue in the customer projects is downtime. Since the customers are using the software during normal work hours, critical tasks cannot be interrupted by version changes. Currently the time slot for version deployment is negotiated with the customer, and the downtime is positioned during times where the software isn't under heavy usage. The experiments must therefore be deployed to production environments such that they don't cause downtime.

The customers have also been trained to perform certain tasks with the software. If the UI constantly undergoes major changes, the customers might get lost. There is already some reluctancy towards new versions, since new versions possibly introduce new bugs and certain UI elements might have their places changed.

The case company's products are mostly installed to customer environments. The typical deployment flow includes a customer testing phase in the user acceptance testing environment before a production version can be deployed. The customers perform UAT only when they are informed that a new version is available. The production environment is seen critical enough that no major bugs are allowed, and thus the UAT testing cycle is seen favorable by both the customer and the company. The speed by which experiments with customers can be conducted is relative to the speed by which production deployments can be made.

In B2B domain the feedback does not always come straight from end users. Feedback relies on what the other companies representatives say, and this often includes their own opinions. Companies want the software for their own use, and they have requirements for property rights, security and others. In Dialog, the customer company's representatives perform marketing operations with the software. Also their customers can be considered as users, since they are the subjects of the marketing operations, and interact with the software by filling web forms. Some of the customer companies have strict rules for which data can be collected. This adds to the legal obstacles, and customer's consent for user behavior collection is required to collect data from their users.

The interviewees state that having a pro-active lead customer is required to both design the experimentation process and to practice it with the customer. The interviewees are unsure how experimenting affects the engagement model with the customer, and what is required from the customer.

D. Continuous Experimentation's benefits to the case company

In the case company, the selection of development ideas is based almost exclusively on the opinions of management of the organization. The ideas for new features are often discussed in the steering group where only management is present. Studies report similar results in other companies as well [11].

The business value of the software product can be increased by focusing the development on features that have the largest impact. The case company exposes features to customers only as soon as they are complete. By utilizing minimum viable features and minimum viable products, the decision to develop to feature or a product can be made from an earlier stage. This enables conserving resources if the feature or product is found to be incompetent, or increasing the resources if the feature is seen as beneficial.

Linking product development and business aspects verifies that the product is moving towards a direction with a higher business value. The benefit of utilizing data for guiding product development is better knowledge of customer behavior, which can be used to better react to customers needs. Conducting experiments also allows companies to quantify business hypotheses and analytically derive answers, effectively enabling the use of hypotheses to steer the development process.

The case company does not yet capitalize measuring metrics as part of the decision making process. By constantly measuring new features of the product, breaking the product can be avoided. Breaking the product essentially means that features diminish the value of the product are implemented. Such features could for example be UI changes that are thought to increase the user experience, but fail to do so due to incorrect presumptions.

Dialog is seen as a suitable candidate for experiments due to the extensive user interface and human users. The important metrics stated by the product owner and management include shortest path to complete a task, and amount of paths available to perform certain features. A path consists of clicks required to perform a single task, such as to create a marketing campaign for a new customer group. However, the metrics are only based on the person's own visions and perspectives.

Continuous experimentation can be practiced with different types of experiments that can guide the development of CDM. Building alternative implementations of existing features with attached instrumentation measuring performance can be used to determine if the current implementation or alternative implementation is preferable.

One of the main benefits of continuous experimentation in the case of CDM is increasing the knowledge of how customers are using the system. With the current development, user feedback or usage data is not systematically collected and capitalized. While the customers do provide feedback upon version releases, these happen relatively infrequently and a lot of development resources have already been spent on the features. Also a possibility is to focus the development towards important errors based on the usage data.

E. Systematically introducing Continuous Experimentation

Based on the previous research questions, the following steps were identified as critical requirements in the case company's transition towards continuous experimentation.

- Finding a pro-active lead customer
- Allocating time and resources for product development
- Implementing an infrastructure for experimentation
- Identifying metrics in the software products that increase the value for customer
- Investigating required legal agreements associated with data collection
- Educating employees to increase the competence in experimentation

The pro-active lead customer allows for exploring the new engagement model. The company has to implement the infrastructure to support experimentation. Identifying which aspects of the product are valuable to the customer and the metrics to measure these attributes is also required to lead the experimentation design.

Fagerholm et al. define that a suitable experimentation system must be able to release minimum viable products or features with suitable instrumentation for data collection. The experimentation system must also allow the design and management of experiment plans, linking experiment results with a product roadmap and managing a flexible business strategy [12]. Next, the continuous experimentation model and Build-Measure-Learn cycle [12] are analyzed in the context of the case company in three phases of experimentation: design, execute and analyze. The process as described in [12] is refined to address to the challenges found in RQ1.

The original model starts with forming an assumption from the business strategy. A hypothesis based on the assumption is then created, and a minimum viable product (MVP) or a minimum viable feature (MVF) is built with tools for data collection. The MVP or MVF is then used for experiments. In an experiment, the data from the MVF or MVP is collected, analyzed and measured in order to validate the hypothesis. A decision is then made to either persevere with the current strategy or to pivot into a different direction. The refined version presents step-by-step instructions for continuous experimentation in the context of the case company.

1) Design:

1) Form an assumption from the business strategy that is thought to increase the value of the product
2) Form a hypothesis based on the assumption that can be subject to experimental testing
3) Define the type of the experiment and the experimental problem so that it can be run with trustworthy results

4) If running a controlled experiment, define an Overall Evaluation Criteria that can be collected and used to provide an answer to the hypothesis
5) Implement the MVF or MVP
6) Implement the instrumentation to collect the metric

The process of designing an experiments starts by forming an assumption from the business strategy, which is perceived as increasing the value of the product for the customer. After the hypothesis is formed, the type of experiment to test the hypothesis needs to be chosen. The experiment must be able to provide worthful results, and for experiments relying on statistical importance, the amount of users of the software has to be sufficient. For experiments relying on user behavior, statistical importance might not be necessary to draw results.

Then, if running a controlled experiment, the overall evaluation criteria (OEC) is explicitly defined before the feature or MVP is implemented. The OEC is a metric that the company aims to improve with the experiment. One of the common pitfalls in running controlled experiments is to pick an OEC for which the control group is easy to beat by doing something "wrong" from a business perspective [17]. Picking the OEC also concretizes the experiment, making the purpose and goal of the experiment more obvious.

After the OEC has been formed and the experiment type decided, the minimum viable feature (MVF) or minimum viable product (MVP) is implemented, and the instrumentation for data collection is added to the product. The suitable data collection instrumentations for the case company's product according to the developers is a server-side service layer, which can be used to log data from multiple services. For this phase, the technical infrastructure has to be built that supports quickly designing, executing and analysing experiments.

The project management tools used by the teams suit the experimentational model. A kanban board is already used to transfer tasks from the management to the developers, and can be used by to develop experiments as well. The product owners and analysts must first design the experiment, and a developer then checkouts the task and implements the MVF or MVP with instrumentation.

2) Execute:

1) Deploy the version to user acceptance testing environment
2) Perform acceptance testing in the user acceptance testing environment if necessary, and negotiate the production deployment
3) Deploy the version to production environment
4) Run the version for a period of time, and collect the data into a database

In the execute phase the product is deployed to an environment with the experimental feature and instrumentation for data collection. For the case company, the deployment has to undergo testing in the UAT environment. The UAT environment is also suitable for testing the experiment itself and the impact of the experimental feature on the product. When the UAT phase is complete, the production deployment is negotiated and performed. With automated deployments, the production deployment should happen by a click of a button. The software is the run for a period of time, while the required data is collected into a database located in the environment.

3) Analyze:

1) Upload the data from the database and analyze it through the infrastructure
2) Draw conclusions from the data to validate the hypothesis. Based on the validity of the initial assumption, make a decision whether to develop the new feature or product further, keep it as it is, or to cease it and revert back to the unmodified version.

In the analysis phase the results from the experiment are collected from the database and analyzed through analytics software. The business strategy is then persevered or pivoted based on whether the hypothesis is valid and supports the assumption. For this phase, the only deviation is that the data has to be uploaded from the customer's environment into the analyze environment.

V. Conclusion

The study investigated extending the development process towards continuous experimentation to advance the company towards real-time value delivery. The research questions address the challenges faced in this transition, the benefits found from this practice and systematical application of continuous experimentation in the B2B domain.

This study has identified the main requirements a company operating in the B2B domain has to address when applying continuous experimentation. Figure 1 visualizes the challenges in the three different aspects. These challenges are mostly caused by having multiple customers that rely on the software. Because the software product is important for the customer's business, major changes that degrade the user experience should be avoided. Since each customers business has unique properties, experiments that provide value for a single customer might not provide as much value for another customer. Also, the speed by which experiments with customers can be conducted is relative to the speed by which production deployments can be made. Additionally, moving towards continuous experimentation requires an organizational culture shift towards an experimental mindset.

The benefits of this practice matched to many problems found in the case company, and a company operating in similar domain with similar products can use them as a basis when considering applying this practice. Continuous experimentation allows the case company to better guide its product development, and rely on quantitative or qualitative data instead of opinions when choosing which features or products to develop further. By utilizing the minimum viable features or minimum viable products, the case company can quickly expose new features to the customer and receive feedback in real-time.

The study identified the main objectives the case company needs to achieve to apply continuous experimentation. The deviations found in the Fagerholm et al. model [12] included

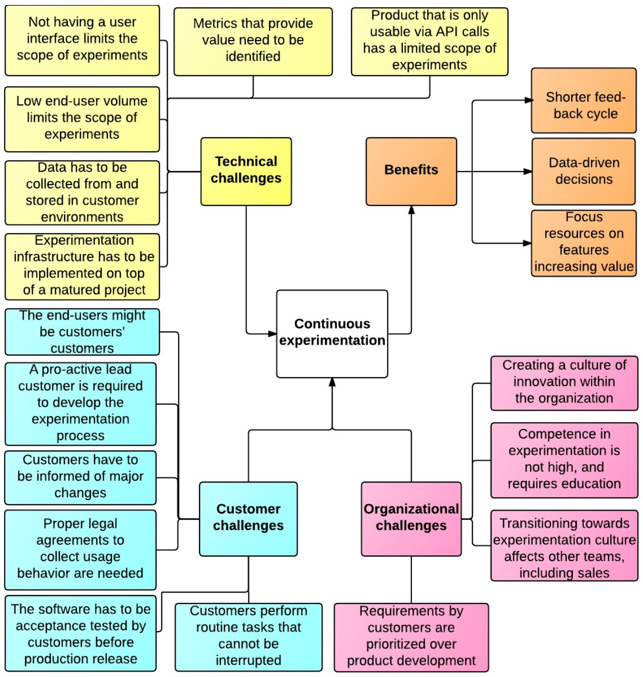
Not having a user interface limits the scope of experiments
Metrics that provide value need to be identified
Product that is only usable via API calls has a limited scope of experiments
Low end-user volume limits the scope of experiments
Data has to be collected from and stored in customer environments
Experimentation infrastructure has to be implemented on top of a matured project
Technical challenges
Benefits
Shorter feed-back cycle
Data-driven decisions
Focus resources on features increasing value
Continuous experimentation
The end-users might be customers' customers
A pro-active lead customer is required to develop the experimentation process
Customers have to be informed of major changes
Proper legal agreements to collect usage behavior are needed
The software has to be acceptance tested by customers before production release
Customer challenges
Organizational challenges
Creating a culture of innovation within the organization
Competence in experimentation is not high, and requires education
Transitioning towards experimentation culture affects other teams, including sales
Customers perform routine tasks that cannot be interrupted
Requirements by customers are prioritized over product development

Fig. 1. Challenges and benefits of Continuous Experimentation.

deploying versions to user acceptance testing environment before a production release is made, and defining an overall evaluation criteria for controlled experiments.

A. Limitations

Since case studies only allow analytic generalisations instead of statistical generalisations, the findings cannot be directly generalized to other cases. However, the phenomena was deeply understood through gathering a large amount of qualitative data and systematically analysing it. Therefore the core findings should be applicable to similar research problems outside of the empirical context of this study. This means that the B2B challenges and benefits of continuous experimentation can be considered as a starting point for further studies in other contexts where this development model takes place.

B. Future work

An interesting question regarding the continuous experimentation model discussed in RQ3 is how to build an effective back-end infrastructure for experimentation. Additionally, a possible avenue for further research is to identify to what extent the core findings of this study can be generalized to other companies working in the B2B domain with different software products.

C. Acknowledgements

We wish to thank the case company for participating in the study and the reviewers for their highly valuable comments. We would also like to thank the Finnish technology agency, Tekes, for funding the Cloud Software Factory project, and the Need for Speed program, under which the proposed study was undertaken. This paper is based on thesis work [18] completed at the University of Helsinki.

References

[1] Olsson, H. H., Alahyari, H., & Bosch, J. (2012, September). 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.
[2] Neely, S., & Stolt, S. (2013, August). Continuous Delivery? Easy! Just Change Everything (well, maybe it is not that easy). In Agile Conference (AGILE), 2013 (pp. 121-128). IEEE.
[3] Bosch, J. (2012). Building products as innovation experiment systems. In Software Business (pp. 27-39). Springer Berlin Heidelberg.
[4] McKinley, D. Design for Continuous Experimentation. http://mcfunley.com/design-for-continuous-experimentation (January, 2015)
[5] Lindgren, E., Münch, J. (2015). Software Development as an Experiment System: A Qualitative Survey on the State of the Practice. In Proceedings of the 16th International Conference on Agile Software Development (XP 2015), LNBIP
[6] Eklund, U., & Bosch, J. (2012, August). Architecture for large-scale innovation experiment systems. In Software Architecture (WICSA) and European Conference on Software Architecture (ECSA), 2012 Joint Working IEEE/IFIP Conference on (pp. 244-248). IEEE.
[7] King, N. (1998) Template analysis. In Qualitative methods and analysis in organizational research: A practical guide (pp. 118-134). Sage Publications Ltd.
[8] Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. Empirical software engineering, 14(2), 131-164.
[9] Dzamashvili Fogelström, N., Gorschek, T., Svahnberg, M., & Olsson, P. (2010). The impact of agile principles on marketdriven software product development. Journal of Software Maintenance and Evolution: Research and Practice, 22(1), 53-80.
[10] Loshin, D. (2010). Master data management. Morgan Kaufmann.
[11] Kohavi, R., Henne, R. M., & Sommerfield, D. (2007, August). Practical guide to controlled experiments on the web: listen to your customers not to the hippo. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 959-967). ACM.
[12] Fagerholm, F., Guinea, A. S., Mäenpää, H., & Münch, J. (2014, June). Building blocks for continuous experimentation. In Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering (RCoSE 2014), Hyderabad, India.
[13] Ries, E. (2011). The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses. Random House LLC.
[14] Kaushik, A. Experimentation and Testing: A Primer. http://www.kaushik.net/avinash/experimentation-and-testing-a-primer (January, 2015)
[15] Moran, M. (2007). Do it wrong quickly: how the web changes the old marketing rules. IBM Press.
[16] Kohavi, R., Deng, A., Frasca, B., Walker, T., Xu, Y., & Pohlmann, N. (2013, August). Online controlled experiments at large scale. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 1168-1176). ACM.
[17] Crook, T., Frasca, B., Kohavi, R., & Longbotham, R. (2009, June). Seven pitfalls to avoid when running controlled experiments on the web. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 1105-1114). ACM.
[18] Rissanen, O., Münch, J. (supervisor), Männistö, T. (supervisor). Extending the Development Process Towards Continuous Delivery and Continuous Experimentation in the B2B Domain: A Case Study. Master's Thesis. University of Helsinki (2015)
[19] Olsson, H. H., & Bosch, J. (2014, August). From Opinions to Data-Driven Software R&D: A Multi-case Study on How to Close the 'Open Loop' Problem. In Software Engineering and Advanced Applications (SEAA), 2014 40th EUROMICRO Conference on (pp. 9-16). IEEE.