The Evolution of Continuous Experimentation in Software Product Development

From Data to a Data-driven Organization at Scale

Aleksander Fabijan[1], Pavel Dmitriev[2], Helena Holmström Olsson[1], Jan Bosch[3]

[1]Malmö University
Faculty of Technology and Society
Malmö, Sweden
aleksander.fabijan@mah.se
helena.holmstrom.olsson@mah.se

[2]Microsoft
Analysis & Experimentation
Microsoft, One Microsoft Way,
Redmond, WA 98052, USA
padmitri@microsoft.com

[3]Chalmers University of Technology
Dep. of Computer Science & Eng.
Göteborg, Sweden
jan.bosch@chalmers.se

***Abstract*—Software development companies are increasingly aiming to become data-driven by trying to continuously experiment with the products used by their customers. Although familiar with the competitive edge that the A/B testing technology delivers, they seldom succeed in evolving and adopting the methodology. In this paper, and based on an exhaustive and collaborative case study research in a large software-intense company with highly developed experimentation culture, we present the evolution process of moving from ad-hoc customer data analysis towards continuous controlled experimentation at scale. Our main contribution is the "Experimentation Evolution Model" in which we detail three phases of evolution: technical, organizational and business evolution. With our contribution, we aim to provide guidance to practitioners on how to develop and scale continuous experimentation in software organizations with the purpose of becoming data-driven at scale.**

A/B testing; continuous experimentation; data science; customer feedback; continuous product innovation; Experimentation Evolution Model; product value; Experiment Owner

I. INTRODUCTION

Software development organizations and their product development teams are increasingly using customer and product data to support decisions throughout the product lifecycle [1], [2]. Data-driven companies acquire, process, and leverage data in order to create efficiencies, iterate on and develop new products, and navigate the competitive landscape [1]. Digitally adept and technology driven companies are as much as 26 percent more profitable than their competitors [3]. Recent software engineering research reflects this situation with a number of publications on how to change and efficiently conduct controlled experiments to become data-driven [4], [5], [6], [7], [8], [27]. The role of data scientists is increasingly gaining momentum in large software companies [9]. However, despite having data, the number of companies that efficiently use it and that successfully transform into data-driven organizations stays low and how this transformation is done in practice is little studied [10], [11].

In this paper, we present the phases that teams at Microsoft evolved through in order to become data-driven at scale by establishing a controlled experimentation platform and a data-driven mindset. The impact of scaling out the experimentation platform across Microsoft is in hundreds of millions of dollars of additional revenue annually. The journey from a company with data to a data-driven company, however, was not a jump but rather an evolution over a period of years. This development occurs through phases and we illustrate this process by creating the "Experimentation Evolution Model". With this model, we describe the steps to take while evolving data-driven development practices towards continuous experimentation at scale. With our contribution, we aim to provide guidance to practitioners on how to develop and scale continuous experimentation in software organizations and thus become truly data-driven.

The paper is organized as follows. In Section II we present the background and the motivation for this study. In section III, we describe our research method, the data collection and analysis practices and our case company. Our empirical findings are in section IV. In section V, we present our main contribution - the "Experimentation Evolution Model". Finally, we conclude the paper in section VI.

II. BACKGROUND

Rapid delivery of value to customers is one of the core priorities of software companies [8]. With this goal in mind, companies typically evolve their development practices. At first, they inherit the Agile principles within the development part of the organization [12] and expand them to other departments [13]. Next, companies focus on various lean concepts such as eliminating waste [14], removing constraints in the development pipeline [15] and advancing towards continuous integration [16] and continuous deployment of software functionality [10]. Continuous deployment, however, is characterized by a bidirectional channel that enables companies not only to send data to their customers to rapidly prototype with them [17], but also to receive feedback data from products in the field. The intuition of software development companies on customer preferences can be wrong as much as 90% of the time [18], [19], [20]. The actual product usage data has the potential to make the prioritization process in product development more accurate as it focuses on what customers do rather than what they say [21], [22]. Controlled experimentation is becoming the norm in advanced software companies for reliably

evaluating ideas with customers in order to correctly prioritize product development activities [4] [5], [6], [7], [8].

A. Controlled Experiments

In a controlled experiment, users are randomly divided between the variants (e.g., the two different designs of a product interface) in a persistent manner (a user receives the same experience multiple times). Users' interactions with the product are instrumented and key metrics are computed [4], [23]. One of the key challenges with metrics is to decide on which to include in an Overall Evaluation Criteria (OEC). An OEC is a quantitative measure of a controlled experiment's objective [24] and steers the direction of the business development. In controlled experimentation, it is intuitive to measure the short-term effect, i.e., the impact observed during the experiment [25]. Providing more weight to advertisement metrics, for example, makes businesses more profitable in the short-term. However, the short-term effect is not always predictive of the long-term effect and consequently should not be the sole component of an OEC [26]. Defining an OEC is not trivial and should be conducted with great care. Kohavi et al. [4], [26], [27] in their papers present common pitfalls in the process of establishing a controlled experimentation system and guidance on how to reliably define an OEC.

Research contributions with practical guides on how to develop an experimentation system have previously been published both by Microsoft [27], [28] and Google [29]. The Return on Investment (ROI) of controlled experimentation has been discussed a number of times in the literature [23], [27]. However, the count of companies that successfully developed an experimentation culture and became data-driven remains low and limited to other web service companies such as Facebook, Google, Booking, Amazon, LinkedIn, Etsy, Skyscanner [10], [28]. We believe that the reason for this unsuccessful adoption of continuous experimentation resides in the lack of knowledge on how the transition can be done in practice. Companies have the necessary instrumentation in place [30], are able to gather and analyze product data, but they fail to efficiently utilize it and learn from it [11].

The research contributions from Google and Microsoft provide guidance on how to start developing the experimentation platform. However, they do not provide guidance on which R&D activities to prioritize in order to incrementally scale the experimentation across the organization. This technical research contribution is aiming to address this gap and provide guidance on how to evolve from a company with data to a data-driven company. We focus on technical challenges (e.g. the necessary platform features that are required for successful scaling) as well as the organizational aspects (e.g. how to integrate data scientists in product teams) and business aspects (e.g. how to develop an Overall Evaluation Criteria). This leads to the following research question:

RQ: "*How to evolve controlled experimentation in software-intensive companies in order to become data-driven at scale?"*

To address this research question, we conducted a mixed methods study of how continuous experimentation scaled at Microsoft. We describe the research method in detail next.

III. METHOD

This research work is an inductive case study and was conducted in collaboration with the Analysis and Experimentation (A&E) team at Microsoft. The inspiration for the study originates from an internal model used at A&E, which is used to illustrate and compare progress of different product teams on their path towards data-driven development at scale. The study is based on historical data points that were collected over a period of two years and complemented with a series of semi-structured interviews, observations, and meeting participations. In principle, it is an in-depth and single case study [31], however, our participants are from different organizational units and product teams with fundamentally different product and service offerings. Several of the participants worked in other data-driven organizations before joining the Microsoft A&E team. The A&E team provides a platform and service for running controlled experiments for customers. Its data scientists, engineers and program managers are involved with partner teams and departments across Microsoft on a daily basis. The participants involved in this research work are primarily collaborating with the following Microsoft product and services teams: Bing, Cortana, Office, MSN.com, Skype and Xbox.

A. Data Collection

The data collection for this research was conducted in two streams. The first stream consisted of collection of archival data on past controlled experiments conducted at Microsoft. The first author of this paper worked with the Microsoft Analysis & Experimentation team for a period of 10 weeks. During this time, he collected documents, presentations, meeting minutes and other notes available to Microsoft employees about the past controlled experiments, the development of the experimentation platform and organizational developments conducted at Microsoft A&E over the last 5 years. In cumulative, we collected approximately 130 pages of qualitative data (including a number of figures and illustrations).

The second stream consisted of three parts. The first author (1) participated in weekly experimentation meetings, (2) attended internal training on controlled experimentation and other related topics, and (3) conducted a number of semi-structured interviews with Microsoft employees. In all three data collection activities, the first author was accompanied by one of the other three authors (as schedules permitted). At all meetings and training, we took notes that were shared between us at the end of each activity. The individual interviews were recorded and transcribed by the first researcher.

The second author of this paper has been working with the Analysis & Experimentation team at Microsoft for a period of six years. He was the main contact person for the other three researchers throughout the data collection and analysis period and advised the diverse selection of data

scientist, managers and software engineers that we interviewed. In total, we conducted 14 semi-structured interviews (1 woman, 13 men) using a questionnaire guide with 11 open-ended questions. The participants that work with different product teams were invited for a half an hour interview by the first two authors. The interview format started with an introduction and a short explanation of the research being conducted. Participants were then asked on their experience with conducting controlled experiments, how they document learnings from those experiments, and how their practices changed over time. We also asked for examples of successes, pitfalls, and pain points that they experience while conducting controlled experiments.

We provide a detailed list of our interviewees, their roles and their primary product teams in Table 1 below. The ones with n/a do not collaborate with product teams directly, but are rather focusing on platform development and other activities within the A&E team.

TABLE I. INTERVIEW PARTICIPANTS

	Interview details		
	Role	***Length (min)***	***Product***
1	Senior Data Scientist	45	Skype
2	Data Scientist	45	Skype
3	Principal Group Engineering Mgr.	30	n/a
4	Principal Data Scientist	30	Bing
5	Senior Software Engineer	30	n/a
6	Senior Data Scientist	45	MSN
7	Principal Data Scientist Mgr.	30	Office
8	Principal Data Scientist Mgr.	30	Office
9	Principal Data Scientist & Architect	30	Bing
10	GPM Program Manager	30	n/a
11	Principal Software Engineer	30	Bing
12	Senior Applied Researcher	30	Ads
13	Senior Program Manager	30	Bing
14	Senior Program Manager	30	Cortana

B. Data Analysis

We analyzed the collected data in two steps. First, we grouped the data that belonged to a certain product. Next, we grouped products in 4 buckets based on the number of experiments that their product teams are capable of executing per annum (i.e. 1-9, 10-99, 100-999, and 1000+). Second, and with the goal to model the evolution of continuous experimentation, we performed inductive category development [32]. In the first step, we emerged with three high level definitions of categories that represent our research interest (namely technical evolution, organizational evolution and business evolution). Next, we formulated the categories under each of the three categories by reading through the collected data and assigning codes to concepts that appeared in it. This approach is similar to the Grounded Theory approach as we didn't have preconceptions on which categories to form beforehand [33]. The final categories are visible in our model in Figure 5. To develop the content of the table, we backtracked the codes within the buckets. Using a 'venting' method, i.e. a process whereby interpretations are continuously discussed with professional colleagues, we iteratively verified and updated our theory on the content for each of the four phases of our models in Figure 5. The A&E team provided continuous feedback on the developing theory and helped to clear any discrepancies in the raw data.

C. Validity Considerations

1) Construct Validity

To improve the study's construct validity, we complemented the archival data collection activities with individual semi-structured interviews, meetings and trainings. This enabled us to ask clarifying questions, prevent misinterpretations, and study the phenomena from different angles. Meeting minutes and interview transcriptions were independently assessed by three researchers to guarantee inter-rater reliability. Since this study has been conducted in a highly data-driven company, all the participants were familiar with the research topic and expectations between the researchers and participants were well aligned. The constructed artifact was continuously validated with the A&E team members during the study.

2) External Validity

The main result of this paper details an evolution towards becoming a data-driven company as experienced at Microsoft. The first author conducted this research while collaborating with the second author who is permanently employed at the case company. This set-up enabled continuous access and insight. However, and since this approach differs from a traditional case study [31], the contributions of this paper risk being biased from this extensive inside view. The main contribution can thus not directly translate to other companies. However, we believe that the phases of our model, especially the dimension concerning the technical evolution, are similar to the ones that other software companies traverse on their path towards becoming data-driven. The 'Experimentation Evolution Model' can be used to compare other companies and advise them on what to focus on next in order to efficiently scale their data-driven practices. The embedded systems domain is one example area where companies are aiming to become data-driven and that we previously studied [34], [11], [35]. The phases of our model can be applied to this domain.

In the next section, we show the empirical data by describing four controlled experiments from different product teams.

IV. EMPIRICAL FOUNDATION

In this section, we briefly present examples of controlled experiments conducted at Microsoft. The space limitations make it difficult to show all the depth and breadth of our

empirical data. Due to this limitation, we select four example experiments. With each of them, we aim to illustrate the capabilities and limitations that product teams at Microsoft experience as they evolve their data-driven practices. We start with Office, where data-driven development is beginning to gain momentum and where the first controlled experiments were recently conducted. Next, we present an example from Xbox and an example from MSN where the experimentation is well established. Finally, we conclude the section by providing an illustrative experiment from Bing where experimentation is indispensable and deeply embedded in the teams' development process.

A. *Office Contextual Bar Experiment*

Microsoft Office is a well-known suite of products designed for increasing work productivity. Data-driven practices in Office product teams are in the early stages. The product team responsible for the edit interface in Office mobile apps recently conducted a design experiment on their Word, Excel, and PowerPoint apps. They believed that introducing a Contextual Command Bar (see Figure 1 below) would increase the engagement compared to a version of the product without the contextual bar. Their hypothesis was that mobile phone users will do more editing on the phone because the contextual command bar will improve editing efficiency and will result in increased commonality and frequency of edits and 2-week retention.

Figure 1. The "Contextual Bar" experiment on Word mobile app.

During the set-up of the experiment, the team ran into issues with measuring the number of edits. The instrumentation was still in the early stages, and the telemetry teams did not accurately log the edit events. These issues had to be fixed prior to the start of the experiment. The results of a two-week experiment indicated a substantial increase in engagement (counts of edits), but no statistically significant change in 2-week retention. The experiment provided the team with two key learnings: (1) Proper instrumentation of existing features and the new feature is essential for computing experiment metrics, (2) It is important to define global metrics that are good leading indicators and that can change in a reasonable timeframe.

B. *Xbox Deals for Gold Members*

Xbox is a well-known platform for video gaming. Experimentation is becoming well established with this product and their teams have been conducting experiments on several different features.

In one of the experiments, a product team at Xbox aimed to identify whether showing prices (original price and the discount) in the weekly deals stripe, and using algorithmic as opposed to editorial ordering of the items in the stripe impacts engagement and purchases. They experimented with two different variants. On Figure 2, we illustrate the experiment control (A) and both of the treatments (B, C).

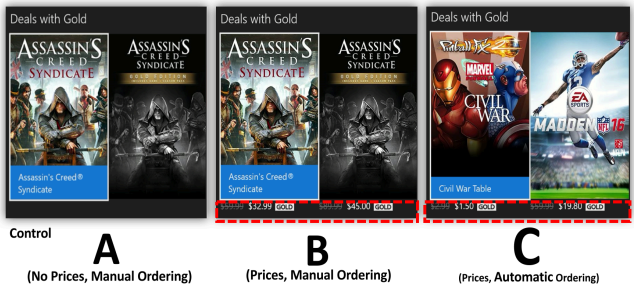

Figure 2. The "Xbox deals" experiment.

At Xbox, instrumentation is well established and a reliable pipeline for data collection exists. Metrics that measure user engagement and purchases are established and consist of a combination of different signals from the logs aggregated per user, session and other analysis units. In contrast to the Office Word experiment above, the Xbox team autonomously set-up their experiments, however, they still require assistance on the execution and monitoring of the experiment and at the analysis stage to interpret results. The two-week experiment showed that, compared to control, treatment B decreased engagement with the stripe. The purchases, however, did not decrease. By showing prices upfront treatment B provided better user experience by engaging the users who are interested in a purchase and sparing a click for those not interested. Treatment C provided even greater benefit, increasing both engagement with the stripe and purchases made. In this experiment the team learned that (1) Showing prices upfront results in better user experience, and (2) Algorithmic ordering of deals beats manual editorial ordering.

C. *MSN.com News Personalization*

In contrast to Office Word and Xbox where experimentation is primarily conducted with features focusing on design changes, teams at MSN.com experiment with most feature changes. In one of the recent experiments, they aimed to test a personalization algorithm developed within Microsoft Research for their news page. The hypothesis was that user engagement with the version that uses the machine learning personalization algorithm would increase in comparison to the manually curated articles. In contrast to Word and Xbox teams, the MSN product team autonomously set-up and execute experiments. A number of