Continuous Experimentation and A/B Testing: A Mapping Study

Rasmus Ros
rasmus.ros@cs.lth.se
Lund University
Sweden

Per Runeson
per.runeson@cs.lth.se
Lund University
Sweden

ABSTRACT

Background. Continuous experimentation (CE) has recently emerged as an established industry practice and as a research subject. Our aim is to study the application of CE and A/B testing in various industrial contexts. **Objective.** We wanted to investigate whether CE is used in different sectors of industry, by how it is reported in academic studies. We also wanted to explore the main topics researched to give an overview of the subject and discuss future research directions. **Method.** We performed a systematic mapping study of the published literature and included 62 papers, using a combination of database search and snowballing. **Results.** Most reported software experiments are done online and with software delivered as a service, although varied exemptions exist for e.g., financial software and games. The most frequently researched topics are challenges to conduct experiments and statistical methods for software experiments. **Conclusions.** The software engineering research on CE is still in its infancy. There are future research opportunities in evaluation research of technical topics and investigations of ethical experimentation. We conclude that the included studies show that A/B testing is applicable to a diversity of software and organisations.

CCS CONCEPTS

•**General and reference → Surveys and overviews;**

KEYWORDS

Continuous experimentation, A/B testing, Mapping study

ACM Reference format:
Rasmus Ros and Per Runeson. 2018. Continuous Experimentation and A/B Testing: A Mapping Study. In *Proceedings of IEEE/ACM 4th International Workshop on Rapid Continuous Software Engineering , Gothenburg, Sweden, May 29, 2018 (RCoSE'18),* 7 pages.
DOI: 10.1145/3194760.3194766

1 INTRODUCTION

The idea of getting continuous feedback from users on the evolution of software, is appealing. This is done with controlled experiments, known as A/B testing in the basic case of comparing two variants, A and B, which are the control and test variable in an experiment. By running multiple variants in an experimental setting with real users, and systematically measure the outcome, developers can make evidence-based decisions to guide their software evolution. Especially for service-based software with large user volumes and centralized software upgrades, design decisions can be underpinned by statistically significant analyses, using experiments. Large internet companies started applying this practice a decade ago, and some of them have published their insights in research venues, among those Microsoft [24], Facebook [2], Google [29], Intuit [4], and Yandex [19].

Conducting experiments in iterations is known as continuous experimentation (CE)—a general term which covers a wide variety of experiments and the implications of experiments on the whole software engineering process. This is an extension to the introduction of continuous integration and continuous deployment, all summarized as continuous software engineering [14]. Efficient delivery of software, through continuous delivery, enables experimentation even on small software changes [18]. The term CE was popularized in software engineering by Fagerholm et al. [13].

Now continuous experimentation is established as a software engineering practice, although there is still a lot to research about its specifics and generality. Publication venues are both in the fields of data science (DS) and software engineering (SE). As with any emergent research field, the terminology is somewhat different between the two, as well as within the fields.

To get an overview, both of the research being conducted, and the practice being studied and reported as research and experience reports on CE, we conduct a systematic mapping study [21]. Our goal is to study the application of CE and A/B testing in industrial contexts. Our aim is to use the study as a basis for future work, identifying results to build on, and gaps in the research to fill [20].

Following this reasoning we formulate three research questions within the context of the studies included in the literature search. **RQ1** What are the main topics researched within CE and how are they studied? **RQ2** Which kind of organisations use CE and what sectors do they operate in? **RQ3** With what type of experiments have CE been applied?

2 RELATED WORK

In the process of our search we did not find any related work of literature studies that focus only on CE. Yaman et al. [32] have conducted a systematic literature review (SLR) of customer involvement of users in continuous deployment. Their main finding is that there are miss-perceptions among customers, which is not relevant for A/B testing where users are minimally aware of the experiment. They also point out that the area remains unexplored academically, but is common in industry, which we also find in this study.

RCoSE'18, Gothenburg, Sweden

978-1-4503-5745-6/18/05…$15.00
DOI: 10.1145/3194760.3194766

3 RESEARCH METHOD

The aim of a mapping study is to provide an overview of the published literature on a topic, identify missed areas in the research or form the basis for future reviews. In this study, we follow the guidelines by Kitchenham et al. [21].

As starting points of the study we conducted a pilot study and created a research protocol. The pilot study was conducted during 2016, but did not find sufficient material from SE venues to continue at the time (as is clear from the trend in Table 1). The papers from the pilot study were used in this study as a validation set for the search terms.

3.1 Search strategy

Defining a precise search strategy for CE was challenging in this study for three reasons: (i) a conflict in terminology with experiments as a research method, (ii) the non-standardized terminology in an emerging topic, and finally (iii) it is an interdisciplinary topic, with main venues on both DS and SE. For those reasons we used a combination of database search and snowballing in iterations as search strategy, as recommended by Ros et al. [28].

The search string was formed by testing various options which were validated against results from the pilot study. The end result is a disjunction of synonym words for controlled experiments and continuous experimentation. We used word endings that express the process of experimentation rather than a single experiment. This strategy was not successful on IEEE Xplore because of automatic word stemming. We considered using negative keywords, such as software, but failed to find any useful terms. We used search filters to limit the results to SE and related research fields, and to limit the publication between 2007–2017. This is motivated by the results of our pilot study and by the paper from 2007 by Kohavi et al. [24] that introduces the topic.

The resulting search string was as follows: *"online experiments"* OR *"online experimentation"* OR *"controlled experiments"* OR *"controlled experimentation* OR *"a/b testing"* OR *"a/b tests"* OR *"split testing"* OR *"split tests"* OR *"bucket testing"* OR *"bucket tests"* OR *"continuous experimentation"* OR *"experiment systems"*.

We executed our queries on the four databases recommended by Kitchenham et al. [21]: (i) *IEEE Xplore* with 230 hits, (ii) *ACM Digital Library* with 1,240 hits, (iii) *Scopus* with 3,007 hits, and (iv) *ISI Web of Science* with 1,017 hits. Total papers from searching databases was 5,494.

We also performed backwards snowballing of the 62 included studies by screening references in the papers, adding 1,634 hits. Eight additional papers were found through snowballing, and those were also subject to snowballing. The snowballing was performed in conjunction with the selection phase. There was no relevant literature from before 2007 found through snowballing.

3.2 Selection strategy

The primary study selection was performed by the first author in three steps. First, duplicates were removed with the tool Jabref in a semi-automated process, keeping 3,425 of the 5,494 papers found from the database search. The second step was a screening based on: title, abstract, and keywords. In the screening we kept 86 of the 3,425 papers. The removed papers were clearly not relevant; usually because they used controlled experiments as a research method rather than researching controlled experiments as such.

The final step was to read the 86 papers and compare against the inclusion/exclusion criteria. Any uncertainties were resolved in discussions with the second author, and the criteria were updated. The end result sums up to 62 included papers. The full inclusion/exclusion criteria were as follows.

Include paper if:

- The research is based on or is conducted by SE practitioners involved with CE.
- The experiments under study have its data sourced from normal usage of the software in a production environment.

Exclude paper if it is:

- The discussed *experiment* is only in the sense of *testing out new ideas* rather than a scientific method.
- Not written in English.
- Not peer-reviewed or is an extended abstract.
- A duplicate of a paper already included (only the latest version is included).

3.3 Data extraction and classification schema

We used both thematic analysis and a classification schema to extract data. The schema was created in discussions between the authors and evolved during the process of extraction. The data was extracted by the first author and reviewed by the second author in discussions. The extracted data is available as supplementary online material[1].

The process for conducting thematic analysis was as follows. We started by carefully reading all included papers. Paragraphs that described research topic, organisations, and experiments were given a theme of a sentence or two. The themes were then grouped into similar topics.

The extracted data was based on three different quantities: (i) characteristics of the papers, (ii) organisations and software sectors in which the studies are conducted, and (iii) characteristics of the experiments. Only organisations that are mentioned to actually conduct experiments are counted. Note also that we define a one-to-one correspondence between organisations and software sectors. As an example, in a paper by authors from Microsoft they discussed both the search engine Bing and the game console Xbox, in which case it was counted as two separate organisations. For experiments, we included only those that gave sufficient background in terms of experimental design and purpose of the individual experiment. For both organisations and experiments we removed duplicates as best as we could, which was sometimes challenging because of anonymity or lack of detail in descriptions.

3.3.1 Paper characteristics (RQ1). Research topic was thematically analysed as described above. For each paper we selected the major 1–3 topics. We divided the papers into software engineering or data science based on research venue, with the intent to analyse the differences between the fields. *Research approach* was classified based on the schema by Wieringa et al. [30]. The four types observed were: (i) *evaluation research* investigates established practices in industry, (ii) *experience reports* state the authors' personal opinion on a subject, (iii) *proposed solution* papers introduce a novel

[1] http://lup.lub.lu.se/search/ws/files/40009496/extracted_data.csv

Table 1: Paper statistics segmented by research fields software engineering (SE) and data science (DS). The trend sparklines show normalized number of yearly papers. The peak count is underlined and red. The main publication venues are listed with their total publication count.

Field	Total	2007–2017	Peak	Main venues
DS	43		9	KDD (19), WWW (7)
SE	17		7	ICSE (4), ICSOB (3)
Other	2	*User design and computer science.*		
Σ	62		15	

solution to a problem, and finally (iv) *validations* provide an initial investigation of a proposed solution.

3.3.2 Organisation characteristics (RQ2). *Company size* was classified based on number of employees: large ≥ 250, medium < 250, and small < 50. When company size was not listed in the paper, but company name was, we searched public internet databases for number of employees. *Business model* was categorized as one of business to business or consumer. It was either inferred from the text or explicitly mentioned. *Software sectors* (or domains) were also extracted with thematic analysis from the paper text, as described above.

3.3.3 Experiment characteristics (RQ3). Experiments were classified from the descriptions in the papers, based on three dimensions: (i) Overall *goal* of the experiment: gaining knowledge, increasing revenue, or improved user engagement. (ii) Types of *treatment* (or change) introduced in the experiment: visual or algorithmic change, or entirely new features, and (iii) *experimental design*, most typically A/B-test, A/B/n-test or multi-variate tests (MVT). We excluded experiments that were missing any of these data.

4 RESULTS

The results are divided into three parts, research topics, organisation and experiment characteristics. First we give some observations on the state of the literature, with some general trend statistics in Table 1. The total number of unique experiments reported in the 62 papers are 91, and the number of organisations (organisation–sector pairs) is 42. According to the figures, the research in SE started later and is in its initial phase, while the research from DS reached its peak in 2015. A third of the papers are published in the knowledge discovery and data mining conference (KDD), in which the initial paper was also published [24].

There are many papers with the first author from industry: 43 of 62 (70%). Many of the industrial papers use only proprietary datasets to evaluate their methods. We did not find any publicly available datasets or any available code replicating statistical methods or analysis.

4.1 Research topics (RQ1)

The intention with the following topics overview is not to give a full account of all papers, as only the most influential papers in each topic is mentioned. Influence is measured with citation count per year with Google scholar.

Table 2: Number of each combination of research topic and research approach of the included papers. Research approaches [30] are explained in Section 3.3.

			Research approach			
	Research topic	Total	Evaluation research	Experience report	Validation	Proposed solution
1	Experiment process	**7**	■ 5	■ 2	0	0
2	Infrastructure	**10**	■ 5	■ 3	■ 1	■ 1
3	Challenges	**19**	■ 7	■ 10	0	■ 2
4	Benefits	**3**	■ 1	■ 2	0	0
5	Variability management	**5**	0	0	■ 3	■ 2
6	Metrics	**6**	0	■ 3	■ 1	■ 2
7	Statistical methods	**16**	0	■ 1	■ 3	■ 12
8	Design of experiments	**8**	0	■ 2	■ 3	■ 3
9	Domain considerations	**6**	■ 1	■ 2	0	■ 3
10	Ethics	**1**	■ 1	0	0	0

Table 2 shows a summary of the topics and how they are studied. The only difference in research topic between DS and SE fields are in statistical methods; which is an exclusive DS topic. Research approach also differs between the fields, in which all proposed solutions are from DS, while SE instead favoured evaluation research.

4.1.1 Experiment process. There are several descriptive models or reference models and reports that describe the process that practitioners use for conducting their experimentation. They include steps before and after an experiment is done and how experimentation is carried out in iterations. The most influential of the reference models are Experiment Systems [4] and RIGHT [13]. The descriptive models are mostly from industry, such as the process for experiments with recommendation systems at Netflix [1].

4.1.2 Infrastructure. Infrastructure is what is necessary or useful to conduct experimentation: system and software architecture, roles required, and organisational culture. The architecture is needed in contexts in which continuous deployment and multiple system version is challenging, such as mobile or embedded [5]. The papers overlap slightly with *experiment process*, with the RIGHT [13] model discussing roles for experimentation and the Netflix report [1] also discussing organisation culture for experimentation.

4.1.3 Challenges. The most frequent topic is challenges for conducting experiments. We found four types of challenges: (i) *technical challenges* such as rapid and continuous deployment and issues in debugging experiments, (ii) *statistical challenges* with violated assumptions of the statistical methods employed, erroneous use, or misinterpretations, (iii) *management* and *organisational challenges* in adopting experiments, communicating results to stakeholders, etc., and (iv) *business challenges* such as prioritising what to experiment, lack of users, or aligning metrics with business. The challenges have been identified in large scale experimentation at

Microsoft [7], at small start ups [13], and in companies adopting experimentation [25].

4.1.4 *Benefits.*

Many authors mention the benefits of CE only in passing, e.g., in a discussion on motivation [4, 24]. Only one study focuses solely on the benefits: Fabijan et al. [12], claiming that the benefits include knowledge gain about the users and improving the quality of software.

4.1.5 *Variability management.*

A/B testing incurs increased variability—by design—in a software system. This topic deals with solutions in the form of tools and techniques to manage said variability. Techniques such as methods of representing valid configurations and how to map the variables in the configuration to software and experiments. Google have tools to manage overlapping experiments in large scale [29]. Facebook has published an open source framework specialised for configuring experiments [3]. Cámara and Kobsa [6] present a systematic approach using feature models from software product lines.

4.1.6 *Metrics.*

Defining measurement for software is not straightforward if the software is not directly aligned with business (as is the case in e.g., e-commerce). Metrics for ads is a recurring topic, in which revenue generated from ad clicks must be balanced with a regression in user experience quality. Defining a metric that quantifies the long-term effect of this trade off is not trivial. Research on this topic includes ways to measure or evaluate how useful metrics are [9], and strategies for dealing with hundreds of metrics [22].

4.1.7 *Statistical methods.*

The challenges with experimentation motivates improved statistical techniques specialised for A/B testing. Aborting experiments pre-maturely in case of outstanding results is a hotly debated topic on the internet and in academia, under the name of continuous monitoring and early stopping [8, 19]. The reason for wanting to stop early is to reduce opportunity cost and the reason not to is due to issues with multiple testing caused by continuous monitoring (which is not allowed in classic statistics without adjustment). Another category of statistical methods is using pre-experiment data to improve the reliability of the experiment, often by addressing the reportedly inflated false positives rates. This technique is employed by e.g., Microsoft [10] and Google [17].

4.1.8 *Design of experiments* or *automated experimentation.*

As seen in Section 4.3 the most common design of experiments is A/B tests. Kohavi et al. [23] caution against using multi-variate tests (MVT) and complex designs because it is easy to make mistakes when introducing multiple changes in the software at once. In contrast, other researchers take an optimization approach and automate experimentation with techniques such as metaheuristic search (genetic algorithms) [26] or multi-variate bandit optimization [16]. They apply it to as many parameters as possible in software components that can be easily parametrized, such as colors and layout of components in a GUI. Other research directions include quasi-experiments because of the lower technical complexity [15]. A quasi-experiment (or natural experiment) is done sequentially instead of in parallel. Finally, mixed methods research is used to combine quantitative and qualitative data for more nuanced feedback [4].

4.1.9 *Domain considerations.*

Some organisations or software sectors (see also 4.2) have domain specific challenges or techniques for experimentation, of which we found five prominent. (i) For *mobile apps* the challenge is continuous deployment and handling multiple variants in GUI layouts. (ii) *Embedded software* face the same challenges as mobile but to a higher degree. There is no specialised infrastructure for deployment of software as there exists for mobile (app stores). Research of experimentation for embedded software is in an early stage; on how to apply it [5]. (iii) Facebook [2] describes that users of *social media* influence each other across experiment groups (thus violating the independence assumption of statistical tests). (iv) For *search engines*, there are special experimental designs for interleaving content, such that both control and treatment is mixed on a single search engine result page. Finally, (v) in *business to business* (B2B) software, the challenge is often to get access to end-user data from their business customers [27].

4.1.10 *Ethics.*

This topic concerns guidelines and implications for ethical experimentation, which should be based on input from legislation, practitioners, and users. We found only one study on ethics: Yaman et al. [31] surveyed practitioners on the ethics of experiments on users without notification or approval. The only question that practitioners agreed on was that users should be notified if personal information is collected. A discussion workshop at WSDM on ethics of online experimentation deserves a special mention; the only record of it is an extended abstract [11].

4.2 Organisations under study (RQ2)

By studying the organisations that use continuous experimentation we identify under what circumstances it is possible to conduct it. In Table 3, to the left we list the software sectors and technological platform of the 42 organisations identified and to the right we show the business model and company sizes.

Software license model and sector is included to showcase the diversity among the software. The license model is based on how software is consumed: *subscribed or free* (as in *gratis*) usually running in a web browser or another service based delivery mechanism, *perpetual* in which the consumer pays a one time cost and is usually an installed application, or *embedded* where the software comes with a hardware purchase. E-commerce is considered free software because what is purchased is not the software running the store. The distinction of business model as B2B or B2C is whether the company is able to directly affect their revenue through their users. As an example, free-to-use software with ads is classified as B2C because they can increase revenue through more clicks on ads, even though the actual revenue is derived from other businesses, paying for ads placement.

4.3 Experiments under study (RQ3)

To further understand what type of software can be experimented on, we examined the experiments featured in the literature. Only some of them have sufficient details to be listed, namely what treatment is applied, what the overall goal of the experiment is, and what the experimental design is. The 91 experiments are totalled for each category in Table 4.

The data is segmented based on whether the experiment design is simple or complex (definitions of the designs are mentioned in

Table 3: Distribution of the 42 organisations and software sectors in the included studies. Software sector (to the left) is segmented on how software is consumed.

Software sector	
Subscribed or free	**Σ 29**
E-commerce	9
Search engine	5
Social media	3
Company website	3
Media streaming	3
News	2
Other *	4
Perpetual	**Σ 9**
Finance and accounting	4
Gaming	2
Other †	3
Embedded ‡	**Σ 4**

Business model		
Business to consumers	(B2C)	32
Business to business	(B2B)	10

Company size (employees)		
Large	(≥ 250)	29
Medium	(< 250)	5
Small	(< 50)	8

*The other subscribed/free sectors: unspecified, e-mail, fitness tracking, and support site.

†The other perpetual sectors: software development tools, IT solutions, and word processing.

‡Embedded firmware sectors: mobile phones, automotive infotainment, alarms, networking hardware.

Table 4: Distribution of the 91 experiments with different types of treatments and goals (knowledge, revenue, and engagement) featured in the included papers. The data is segmented on number of test groups, with corresponding example experimental designs listed to the far right.

	Goal of experiment			**Example experimental designs**
Treatment	Know.	Rev.	Eng.	
Single treatment				
Visual change	■ 3	■ 16	■ 33	*A/B-test, quasi-experiment.*
Algorithmic change	■ 4	■ 5	■ 14	
New feature	0	0	■ 4	
Multiple treatments				
Visual changes	0	■ 5	■ 7	*A/B/n-test, MVT, metaheuristic search.*
Algorithmic changes	0	0	0	
New features	0	0	0	

Section 4.1.8). Complex designs introduce more than one change. 77 of 91 of the experiments are simple A/B-tests. The other experiments are: 3 A/B/n-tests, 2 quasi experiments, 3 optimizations, and 6 MVTs. There are no complex designs that change anything other than visuals.

There are three types of treatments in this study: The most common treatment is (i) *visual changes*, which includes changes in layout of GUI components and cosmetic changes of e.g., color, font or text. Static content, such as news articles or e-mail campaigns is also counted as visual if there is no automated ranking component. Examples of (ii) *algorithmic changes* is search engine ranking or performance improvements. (iii) *New features* is the least common treatment. New functionality often include both visual and algorithmic changes. From the descriptions of experiments it is often unclear what the control treatment is for new features.

Experiment goal is also grouped into one of three categories: (i) *Knowledge gain* about the users or domain. In an experiment with knowledge gain as the goal, the software does not permanently change. One interesting example from Bing added a delay to the search results to gain knowledge the importance of performance [24]. (ii) *Revenue* as measured in direct improvement of business figures. This is often used in the e-commerce examples, where it can be measured by number of products sold. In other sectors it is not as easily defined and requires complex metrics. The final goal is improved (iii) *engagement* based on usage or user experience of the software, which is by far the most common experiment featured.

5 DISCUSSION

5.1 State of the research (RQ1)

CE is becoming an established industry practice and research subject. CE research was initiated in DS and is emerging as a SE subject. Both DS and SE are applied fields while SE is more specific to a domain. Thus we conjecture that the future research will be more applied than theoretical.

We have identified 10 research topics (see Section 4.1) and analysed which kind of research is conducted for each. No topic has sufficient empirical evidence to constitute the foundation for a systematic review. From our analysis, three of them are supported by some empirical evidence: *experiment process*, *infrastructure*, and *challenges*. These topics are researched by observing and describing the current practice. Of the topics popular in DS, there are two with no evaluation research: *statistical methods* and *design of experiments*. Performing evaluation research of these topics might require an intervention to introduce the techniques in another organisation.

The *statistical methods* and *design of experiments* are developed for a reason and they are reportedly used by the big companies. There is a lingering question to what extent these methods should be used by other organisations involved with experimentation. All of the statistical methods introduced in the papers are only validated on proprietary datasets. Realistically, this is not going to change because of the sensitive nature of the data sets. There are also no code examples supplied for these topics. This hurts the research opportunities in an applied field like SE to replicate such studies. It also hurts SE practitioners without advanced knowledge in statistics. Additional tool support is not sufficient if they are difficult to use. Instead, we believe this calls for evaluation research with an intervention.

Variability management is another technical topic that is lacking in empirical evidence. There is ample opportunity to evaluate tools and techniques for variability management in an industrial setting. We see CE and A/B testing as a suitable application of techniques from software product lines and feature oriented software development. This connection was drawn already in 2009 by Comara and Kobsa [6], using aspect oriented software development for implementation. We can only guess as to why this has not been embraced by industry or academia; maybe the rapidly changing and diverse ecosystem of software development on the web, makes it hard to work with inflexible models and tools. We reason that a more systematic approach than feature toggles could be valuable, especially in software sectors such as embedded, with a higher need for stable software.

We believe that there is a discrepancy in the number of papers in *benefits* compared to *challenges* for two reasons. Firstly, due to the many pre-requisites (corresponding to the challenges) of experimenting. A deficiency in any of the significant challenges dooms the endeavour to failure. Thus, each challenge has a profound effect on experimentation. Secondly, the benefits might be perceived as obvious and not warranted as a research topic. This is unfortunate because this bias hinders a systematic review in answering the question on whether CE is a worthwhile SE practice.

Ethics for experimentation will no longer be a fringe academic topic when the General Data Protection Regulation (GDPR)[2] legislation takes effect in the EU in May 2018. Whether it is another hurdle or a serious game changer remains to be seen. Ultimately it is a judicial question to settle which solutions are sufficient to protect user integrity and data. Regardless, we believe that ethical experimentation (whatever that is) should be a priority for the SE research community. We believe that SE research can help define what ethical experimentation is and to find solutions and guidelines to do experimentation in compliance with legislation.

In summary, we identify a research gap on evaluation research and tool support on technical topics, such as automated experimentation, statistical methods, and variability management. We see a need for more research on the implications and feasibility of ethical experimentation.

5.2 Experimentation context (RQ2 and RQ3)

We wanted to investigate how applicable A/B testing and CE was in industry and in doing so inspected the organisations and experiments featured in the literature (see Sections 4.2 and 4.3).

We find a diverse spread of organisations and software sectors, ranging from e-commerce to embedded software. For company sizes, we suspect that there are many more small and medium organisations (compared to large organisations) than what is seen in Table 3. We hypothesize that the bias is caused by researchers' increased chance of contact with large organisations. As for software sector, we conclude that experimentation is more frequently occurring when software does not have to be installed in desktop computers or embedded in hardware. It is surprising for us that there were relatively many business to business (B2B) companies involved with experimentation. This shows that A/B testing is not only used to directly increase the revenue (as is the case in e-commerce) but also for the purpose of providing a high quality product. While there are studies on the challenges in experimentation in B2B software [27], we would like to see more guidelines on e.g., how to define metrics in this context.

The examples of experimentation in the literature are diverse. There are more visual changes, than algorithmic changes, and more algorithmic changes than new features. This is most likely caused by the difficulty of application of these treatments; visual changes are easy to make and so are easier to try multiple variants of. Further, new features often involve both algorithmic and visual changes. It can also be the case of researcher bias, because visual changes are easy to explain and show as an example in a paper. Goals of improving engagement was the most common ones, and increased revenue was more common than knowledge gain. Our interpretation of the data is that it is hard to define metrics with a direct connection to revenue. As for knowledge goals, they are most likely rare because it does not directly result in improved software.

The experimental designs found were mostly standard A/B tests. For designs with multiple treatments we saw only visual changes applied. We believe this is the case because it is used on changes that are easy to implement by parametrizing and test multiple alternatives of. The question is how applicable the techniques, and especially automated experimentation, are to general software (possibly entirely without a graphical interface).

In summary, we observe diversity in terms of organisation, software, and experiment characteristics. The research is dominated by companies such as Microsoft, Google, and Facebook. By showing examples of experimentation in diverse contexts we show that experiments is not only for web-based software, by large companies, or for organisations with business to consumer models.

5.3 Threats to validity

The results and conclusions we draw from this mapping study could be affected by threats to validity, which are discussed in this section. The general steps taken to increase the validity of the study are to follow the practices recommended by guidelines [21]: conducting a pilot study, writing a research protocol, and performing data validation by the second author.

5.3.1 *Study coverage.*

This is an inter-disciplinary study, which means that there could be different terms for the same thing. Even though we used both snowballing and database search we acknowledge that we could still have missed relevant papers. Study selection is another source of missing studies where we could have removed studies that should be included.

5.3.2 *Topic coverage.*

Some potentially relevant topics or software sectors could have no publications that discuss them yet, those research gaps cannot be identified with a literature study study.

5.3.3 *Conclusion validity.*

The data extraction might be affected by author bias. The topics we extracted could be arranged or named differently which might affect the interpretation of the data.

6 CONCLUSIONS

The goal of this study was to find how applicable experimentation is for various organisations and software. We also wanted to explore the main topics and discuss what merits further research. We acheived this making a systematic mapping of published research.

We found two main research gaps when analyzing the main topics (RQ1). Firstly, a lack of real world evaluation of technical topics proposed by DS researchers. The proposed methods could be useful if they reach a wider audience which we believe they can find through evaluation research by SE researchers. Secondly, we request additional research in the form of guidelines, procedures, and techniques for ethical experimentation motivated by recent legislation (GDPR).

[2] The GDPR will strengthen the data protection laws for all individuals within EU. The law also legislates extra-territorial data export and in doing so affects all businesses dealing with the data of EU citizens. See https://www.eugdpr.org/.

The analysis shows a diversity of organisations (RQ2) involved with experimentation, they are active in different software sectors, with different sizes, and business models. E-commerce and other online software is the most common sector but CE is applied in other sectors too, such as finance and gaming. Experiments are conducted primarily to improve the user engagement through visual changes (RQ3).

We conclude that continuous experimentation can be an active research subject for many years to come since there are many research gaps to fill. For practitioners, we have shown that A/B testing is used in diverse settings and conclude that A/B testing is a widely applicable practice.

ACKNOWLEDGMENTS

We thank reviewer one for the excellent and extensive comments. This work was partially supported by the Wallenberg Artificial Intelligence, Autonomous Systems and Software Program (WASP) funded by Knut and Alice Wallenberg Foundation.

REFERENCES

[1] X. Amatriain. 2013. Beyond Data: From User Information to Business Value Through Personalized Recommendations and Consumer Science. In *Proc. of the 22nd ACM Int. Conf. on Information & Knowledge Management (CIKM)*. 2201–2208. DOI: http://dx.doi.org/10.1145/2505515.2514701

[2] L. Backstrom and J. Kleinberg. 2011. Network Bucket Testing. In *Proc. of the 20th ACM Int. Conf. on World Wide Web (WWW)*. 615–624. DOI: http://dx.doi.org/10.1145/1963405.1963492

[3] E. Bakshy, D. Eckles, and M. S. Bernstein. 2014. Designing and Deploying Online Field Experiments. In *Proc. of the 23rd ACM Int. Conf. on the World Wide Web (WWW)*. 283–292. DOI: http://dx.doi.org/10.1145/2566486.2567967

[4] J. Bosch. 2012. Building Products as Innovation Experiment Systems. In *Proc. of the Int. Conf. on Software Business (ICSOB)*. 27–39. DOI: http://dx.doi.org/10.1007/978-3-642-30746-1_3

[5] J. Bosch and U. Eklund. 2012. Eternal Embedded Software: Towards Innovation Experiment Systems. In *Proc. of the 2nd Int. Symp. on Leveraging Applications of Formal Methods, Verification and Validation*. 19–31. DOI: http://dx.doi.org/10.1007/978-3-642-34026-0_3

[6] J. Cámara and A. Kobsa. 2009. Facilitating Controlled Tests of Website Design Changes: A Systematic Approach. In *Proc. of the Int. Conf. on Web Engineering (ICWE)*. 370–378. DOI: http://dx.doi.org/10.1007/978-3-642-02818-2_30

[7] T. Crook, B. Frasca, R. Kohavi, and R. Longbotham. 2009. Seven Pitfalls to Avoid When Running Controlled Experiments on the Web. In *Proc. of the 15th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*. 1105–1114. DOI: http://dx.doi.org/10.1145/1557019.1557139

[8] A. Deng, J. Lu, and S. Chen. 2016. Continuous Monitoring of A/B Tests Without Pain: Optional Stopping in Bayesian Testing. In *Proc. of the 3rd Int. Conf. on Data Science and Advanced Analytics (DSAA)*. 243–252. DOI: http://dx.doi.org/10.1109/DSAA.2016.33

[9] A. Deng and X. Shi. 2016. Data-driven Metric Development for Online Controlled Experiments: Seven Lessons Learned. In *Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*. 77–86. DOI: http://dx.doi.org/10.1145/2939672.2939700

[10] A. Deng, Y. Xu, R. Kohavi, and Y. Walker. 2013. Improving the Sensitivity of Online Controlled Experiments by Utilizing Pre-experiment Data. In *Proc. of the 6th ACM Int. Conf. on Web Search and Data Mining (WSDM)*. 123–132. DOI: http://dx.doi.org/10.1145/2433396.2433413

[11] F. Diaz and S. Barocas. 2016. WSDM 2016 Workshop on the Ethics of Online Experimentation. In *Proc. of the 9th ACM Int. Conf. on Web Search and Data Mining (WSDM)*. ACM, 695–696. DOI: http://dx.doi.org/10.1145/2835776.2855117

[12] A. Fabijan, P. Dmitriev, H. Holmström Olsson, and J. Bosch. 2017. The Benefits of Controlled Experimentation at Scale. In *Proc. of the 43rd Euromicro Conf. on Software Engineering and Advanced Applications (SEAA)*. IEEE, 18–26. DOI: http://dx.doi.org/10.1109/SEAA.2017.47

[13] F. Fagerholm, A. Sanchez Guinea, H. Mäenpää, and J. Münch. 2017. The RIGHT model for continuous experimentation. *Journal of Systems and Software* 123 (2017), 292–305. DOI: http://dx.doi.org/10.1016/j.jss.2016.03.034

[14] B. Fitzgerald and K.-J. Stol. 2017. Continuous Software Engineering: A Roadmap and Agenda. *Journal of Systems and Software* 123 (2017), 176–189. DOI: http://dx.doi.org/10.1016/j.jss.2015.06.063

[15] D. N. Hill, R. Moakler, A. E. Hubbard, V. Tsemekhman, F. Provost, and K. Tsemekhman. 2015. Measuring Causal Impact of Online Actions via Natural Experiments: Application to Display Advertising. In *Proc. of the 21th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*. 1839–1847. DOI: http://dx.doi.org/10.1145/2783258.2788622

[16] D. N. Hill, H. Nassif, Y. Liu, A. Iyer, and S.V.N. Vishwanathan. 2017. An Efficient Bandit Algorithm for Realtime Multivariate Optimization. In *Proc. of the 23rd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*. 1813–1821. DOI: http://dx.doi.org/10.1145/3097983.3098184

[17] H. Hohnhold, D. O'Brien, and D. Tang. 2015. Focusing on the Long-term: It's Good for Users and Business. In *Proc. of the 21th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*. 1849–1858. DOI: http://dx.doi.org/10.1145/2783258.2788583

[18] T. Karvonen, W. Behutiye, M. Oivo, and P. Kuvaja. 2017. Systematic Literature Review on the Impacts of Agile Release Engineering Practices. *Information and Software Technology* 86 (2017), 87–100. DOI: http://dx.doi.org/10.1016/j.infsof.2017.01.009

[19] E. Kharitonov, A. Vorobev, C. Macdonald, P. Serdyukov, and I. Ounis. 2015. Sequential Testing for Early Stopping of Online Experiments. In *Proc. of the 38th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*. 473–482. DOI: http://dx.doi.org/10.1145/2766462.2767729

[20] B. A. Kitchenham, D. Budgen, and P. Brereton. 2011. Using Mapping Studies as the Basis for Further Research–A Participant-observer Case Study. *Information and Software Technology* 53, 6 (2011), 638–651. DOI: http://dx.doi.org/10.1016/j.infsof.2010.12.011

[21] B. A. Kitchenham, D. Budgen, and P. Brereton. 2015. *Evidence-Based Software Engineering and Systematic Reviews*. CRC Press.

[22] R. Kohavi, A. Deng, B. Frasca, T. Walker, Y. Xu, and N. Pohlmann. 2013. Online Controlled Experiments at Large Scale. In *Proc. of the 19th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*. 1168–1176. DOI: http://dx.doi.org/10.1145/2487575.2488217

[23] R. Kohavi, A. Deng, R. Longbotham, and Y. Xu. 2014. Seven Rules of Thumb for Web Site Experimenters. In *Proc. of the 20th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*. 1857–1866. DOI: http://dx.doi.org/10.1145/2623330.2623341

[24] R. Kohavi, R. Henne D. M., and Sommerfield. 2007. Practical Guide to Controlled Experiments on the Web: Listen to Your Customers not to the HiPPO. In *Proc. of the 13th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*. 959–967. DOI: http://dx.doi.org/10.1145/1281192.1281295

[25] E. Lindgren and J. Münch. 2016. Raising the Odds of Success: The Current State of Experimentation in Product Development. *Information and Software Technology* (2016). DOI: http://dx.doi.org/10.1016/j.infsof.2016.04.008

[26] R. Miikkulainen, N. Iscoe, A. Shagrin, R. Cordell, S. Nazari, C. Schoolland, M. Brundage, J. Epstein, R. Dean, and G. Lamba. 2017. Conversion Rate Optimization Through Evolutionary Computation. In *Proc. of the Genetic and Evolutionary Computation Conf. (GECCO)*. 1193–1199. DOI: http://dx.doi.org/10.1145/3071178.3071312

[27] O. Rissanen and J. Münch. 2015. Continuous Experimentation in the B2B Domain: A Case Study. In *Proc. of the 2nd Int. Workshop on Rapid Continuous Software Engineering (RCoSE)*. IEEE Press, 12–18. DOI: http://dx.doi.org/10.1109/RCoSE.2015.10

[28] R. Ros, E. Bjarnason, and P. Runeson. 2017. A Machine Learning Approach for Semi-Automated Search and Selection in Literature Studies. In *Proc. of the 21st Int. Conf. on Evaluation and Assessment in Software Engineering (EASE)*. 118–127. DOI: http://dx.doi.org/10.1145/3084226.3084243

[29] D. Tang, A. Agarwal, D. O'Brien, and M. Meyer. 2010. Overlapping Experiment Infrastructure: More, Better, Faster Experimentation. In *Proc. of the 16th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*. 17–26. DOI: http://dx.doi.org/10.1145/1835804.1835810

[30] R. Wieringa, N. Maiden, N. Mead, and C. Rolland. 2006. Requirements Engineering Paper Classification and Evaluation Criteria: A Proposal and a Discussion. *Requirements Engineering* 11, 1 (2006), 102–107. DOI: http://dx.doi.org/10.1007/s00766-005-0021-6

[31] S. Yaman, F. Fagerholm, M. Munezero, H. Mäenpää, and T. Männistö. 2017. Notifying and Involving Users in Experimentation: Ethical Perceptions of Software Practitioners. In *2017 ACM/IEEE Int. Symp. on Empirical Software Engineering and Measurement (ESEM)*. 199–204. DOI: http://dx.doi.org/10.1109/ESEM.2017.31

[32] S. Yaman, T. Sauvola, L. Riungu-Kalliosaari, L. Hokkanen, P. Kuvaja, M. Oivo, and T. Männistö. 2016. Customer Involvement in Continuous Deployment: A Systematic Literature Review. In *Proc. of the 22nd Int. Conf. on Requirements Engineering: Foundation for Software Quality (REFSQ)*. 249–265. DOI: http://dx.doi.org/10.1007/978-3-319-30282-9_18