Information and Software Technology 134 (2021) 106551

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Controlled experimentation in continuous experimentation: Knowledge and challenges

Florian Auer [a,*,1], Rasmus Ros [b,*,1], Lukas Kaltenbrunner [a], Per Runeson [b], Michael Felderer [a,c]

[a] University of Innsbruck, Austria
[b] Lund University, Sweden
[c] Blekinge Institute of Technology, Sweden

ARTICLE INFO

Keywords:
Continuous experimentation
Online controlled experiments
A/B testing
Systematic literature review

ABSTRACT

Context: Continuous experimentation and A/B testing is an established industry practice that has been researched for more than 10 years. Our aim is to synthesize the conducted research.
Objective: We wanted to find the core constituents of a framework for continuous experimentation and the solutions that are applied within the field. Finally, we were interested in the challenges and benefits reported of continuous experimentation.
Methods: We applied forward snowballing on a known set of papers and identified a total of 128 relevant papers. Based on this set of papers we performed two qualitative narrative syntheses and a thematic synthesis to answer the research questions.
Results: The framework constituents for continuous experimentation include experimentation processes as well as supportive technical and organizational infrastructure. The solutions found in the literature were synthesized to nine themes, e.g. experiment design, automated experiments, or metric specification. Concerning the challenges of continuous experimentation, the analysis identified cultural, organizational, business, technical, statistical, ethical, and domain-specific challenges. Further, the study concludes that the benefits of experimentation are mostly implicit in the studies.
Conclusion: The research on continuous experimentation has yielded a large body of knowledge on experimentation. The synthesis of published research presented within include recommended infrastructure and experimentation process models, guidelines to mitigate the identified challenges, and what problems the various published solutions solve.

1. Introduction

Deciding which feature to build is a difficult problem for software development organizations. The effect of an idea and its return-on-investment might not be clear before its launch. Moreover, the evaluation of an idea might be expensive. Thus, decisions are based on experience or the opinion of the highest paid person [1]. Similarly difficult is the assessment of technical changes on products. It can be difficult to predict the effect of a change on software quality, as evidenced by the extensive research on e.g. defect prediction [2,3] or software reliability estimation [4]. Moreover, there are cases in which it is not feasible to test for all necessary scenarios, e.g. in all relevant software and hardware combinations.

Continuous experimentation (CE) addresses these problems. It provides a method to derive information about the effect of a change by comparing different variants of the product to the unmodified product (i.e. A/B testing). This is done by exposing different users to different product variants and collecting data about their behavior on the individual variants. Thereafter, the gathered information allows making data-driven decisions and thereby reducing the amount of guesswork in the decision making.

In 2007, Kohavi et al. [1] published an experience report on experimentation at Microsoft and provided guidelines on how to conduct so-called *controlled experiments*. It is the seminal paper about continuous experimentation and thus represents the start of the academic discussion on the topic. Three years later, a talk from the Etsy engineer Dan McKinley [5] gained momentum in the discussion. In the talk, the term *continuous experimentation* was used to describe their experimentation practices. Other large organizations, like Facebook [33]

* Corresponding authors.
E-mail addresses: florian.auer@uibk.ac.at (F. Auer), rasmus.ros@cs.lth.se (R. Ros), lukas.kaltenbrunner@uibk.ac.at (L. Kaltenbrunner), per.runeson@cs.lth.se (P. Runeson), michael.felderer@uibk.ac.at (M. Felderer).
[1] Both authors contributed equally to this work.

https://doi.org/10.1016/j.infsof.2021.106551
Received 13 May 2020; Received in revised form 6 February 2021; Accepted 8 February 2021
Available online 16 February 2021
0950-5849/© 2021 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

and Netflix [34], which adopted data-driven decision making [35], shared their experiences [36] and lessons learned [37] about experimentation over the years with the research community. In addition, researchers from industry as well as academia developed methods, models and optimizations of techniques that advanced the knowledge on experimentation.

After more than ten years of research, numerous work has been published in the field of continuous experimentation, including work on problems like the definition of an experimentation process [38], how to build infrastructure for large-scale experimentation [39,159], how to select or develop metrics [40], or the considerations necessary for various specific application domains [41].

The purpose of this systematic literature review is threefold. First, to synthesize the models suggested by the research community to find characteristics of an essential framework for experimentation. This framework can be used by practitioners to identify elements in their experimentation framework. Second, to synthesize the various technical solutions that have been applied. In this inquiry, we also include to what degree the solutions are validated. Finally, to summarize and categorize the challenges and benefits with continuous experimentation. Based on this the following four research questions are addressed in this work:

RQ1: What are the core constituents of a CE framework?
RQ2: What technical solutions are applied in what phase within CE?
RQ3: What are the challenges with CE?
RQ4: What are the benefits with CE?

The research method of this study is based on two independently conducted mapping studies [6,7]. We extended and validated the studies by cross-examining the included studies. Thereafter, we applied two qualitative narrative syntheses and a thematic synthesis on the resulting set of papers.

In the following Section 2 an overview of continuous experimentation and related software practices is given. Next, Section 3 describes the research method applied and Section 4 presents the results of the research. In Section 5 the findings are discussed. Finally, Section 6 summarizes the research.

2. Background

In this section we present an overview of continuous experimentation and related continuous software engineering practices. Further, we summarize our two previously published mapping studies. For the novice reader, we recommend Fagerholm et al.'s descriptive model of continuous experimentation [38], or Kohavi et al.'s tutorial on controlled experiments [42], which is a more hands on introduction for continuous experimentation.

2.1. Continuous software engineering

In their seminal paper on controlled experiments on the web from 2007, Kohavi et al. [1] explain how the ability to continuously release new software to users is crucial for efficient and continuous experimentation, which is now known as continuous delivery and continuous deployment. Together with continuous integration, these are the three software engineering practices that allow software companies to release software to users rapidly and reliably [8] and are fundamental requirements for continuous experimentation.

Continuous integration entails automatically merging and integrating software from multiple developers. This includes testing and building an artifact, often multiple times per day. *Continuous delivery* is the process by which software is ensured to be always in state to be ready to be deployed to production. Successful implementation of continuous integration and delivery should join the incentives of development and operations teams, such that developers can release often and operations get access to powerful tools. This has introduced the DevOps [9] role in software engineering with responsibility for numerous activities: testing, delivery, maintenance, etc. Finally, with *continuous deployment*, the software changes that successfully make it through continuous integration and continuous delivery can be deployed automatically or with minimal human intervention. Continuous deployment facilitates collection of user feedback through faster release cycles [10,11]. With faster release cycles comes the ability to release smaller changes, the smaller the changes are the easier it becomes to trace feedback to specific changes.

Fitzgerald and Stol [43] describe many more continuous practices that encompass not only development and operations, but also business strategy; among them continuous innovation and continuous experimentation. Experiments are means to tie development, quality assurance, and business together, because experiments provide a causal link between software development, software testing, and actual business value. Holmström Olsson et al. [12] describe how "R&D as an experiment system" is the final step in a process that moves through the continuous practices.

2.2. Continuous experimentation

The process of conducting experiments in a cycle is called *continuous experimentation*. The reasoning is that the results of an experiment often begets further inquires. Whether the original hypothesis was right or wrong, the experimenter learns something either way. This learning can lead to a new hypothesis which is subject to a new experiment. This idea of iterative improvement is known since long from the engineering cycle or from iterative process improvements, as explained in the models Plan-Do-Check-Act [13] or quality process improvement paradigm (QIP) [14]. The term "continuous experimentation" as used by software engineering researchers refers to a holistic approach [43] which spans a whole organization. It considers the whole software life-cycle, from business strategy and planning over development to operations.

Some authors have included many methods of gathering feedback in continuous experimentation [38,44], including qualitative methods and data mining. These methods are not the focus of this work, though they are also valuable forms of feedback [11,15]. For example, qualitative focus groups in person with selected users can be used early in development on sketches or early prototypes. The human–computer interaction research field has studied this extensively – recently under the name of *user experience research* – and it has also been the subject of software engineering literature reviews in combination with agile development [16,17]. In contrast to the qualitative methods, a controlled experiment requires a completed feature before it can be conducted. It is focused on quantitative data, thus cannot easily answer questions on the rationale behind the results, as qualitative methods can. As such, these methods compliment each other, but they are different in terms of methodology, infrastructure, and process. We discuss the qualitative methods through the lens of controlled experimentation in Section 4.2.9.

A *randomized controlled experiment* (or A/B test, bucket test, or split test) is a test of an idea or a *hypothesis* in which variables are systematically changed to isolate the effects. Because the outcome of an experiment is non-deterministic, the experiment is repeated with multiple *subjects*. Each subject is randomly assigned to some of the variable settings. The goal of the experiment is to investigate whether changes in the variables have a causal effect on some output value, usually in order to optimize it. In statistical terminology, the variable that is manipulated is called the independent variable and the output value is called the dependent variable. The effect that changing the independent variables has on the dependent variable can be expressed with a statistical *hypothesis test*. A significance test involves calculating

a p-value[2] and the hypothesis is validated if the p-value is below a given *confidence level*, often 95%. In addition, properly conducting a controlled experiment requires a power calculation[3] to decide experiment duration.

In software engineering, a controlled experiment is often used to validate a new product feature [156], in that case the independent variable is whether a previous baseline feature or the new feature should be used. These are sometimes called control and test group, or the A and B group, in which case the *experiment design* is called an A/B test. In an A/B test, only one variable is changed; other experiment designs are possible [18,42] but rarely used [7]. To optimize software configuration settings is another use of controlled experiments in software engineering [45]. The dependent variable of the experiment is some measurable metric, designed with input from some business or customer needs. If there are multiple metrics involved with the experiment, then an overall evaluation criteria (OEC) [19] can be used, which is the most important metric for deciding on the outcome of the experiment. The subjects of the experiments are usually users, that is, each user provides one or more data points. In some cases the subjects are hardware or software parameters, for example, when testing optimal compiler settings.

The process of continuous experimentation (see Fig. 1) has similarities to the tradition from science in software engineering research [20] and elsewhere [18]. However, we base the following process on the RIGHT model by Fagerholm et al. [38]. There are five main phases of the process. (1) In the *ideation* phase hypotheses are elicited and prioritized. (2) *Implementation* of a minimum viable product or feature (MVP) that fulfill the hypothesis follows. (3) Then, a suitable *experiment design* with an OEC is selected. (4) *Execution* involves release engineers deploying the product into production and operations engineers monitoring the experiment in case something goes wrong. Finally, (5) an *analysis* is conducted with either statistical methods by data scientists or by qualitative methods by user researchers. If the results are satisfactory the feature is included in the product and a new hypothesis is selected so the product can be further refined. Otherwise, a decision must be made if to persevere and continue the process or if a pivot should be made to some other feature or hypothesis. Lastly, the results should be generalized into knowledge so the experience gained can be used to inform future hypotheses and development on other features.

Many of the papers included in this study are on improved analysis methods. One such direction that need additional explanation is *segmentation*. It is used in marketing to create differentiated products for different segments of the market. In the context of experiments it is used to calculate metrics for various slices of the data in, e.g. gender, age groups, or country. Experimentation tools usually perform automated segmentation [39] and can, for example, send out alerts if a change affects a particular user group adversely.

2.3. Previous work

Prior to this literature review, two independent mapping studies [6,46] were conducted by the authors. Although both studies were in the context of continuous experimentation, their objectives differed.

In their mapping study [46], Ros and Runeson provided a short thematic synthesis of the topics in the published research and examined the context of the research in terms of reported organisations and types of experiments that were conducted. They found that there is a diverse spread of organisations of company size, sector, etc. Although, continuous experimentation for software that does not require installation (e.g. websites) was more frequently reported. Concerning the experimentation treatment types, the authors found more reports about visual changes than algorithmic changes. In addition, the least common type of treatment encountered in literature was new features. Finally, it was observed that the standard A/B test was by far the most commonly used experiment design.

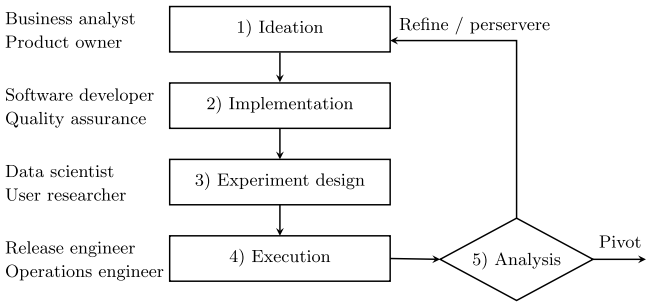

Fig. 1. Continuous experimentation process overview in five phases. A hypothesis is prioritized and implemented as a minimum viable product, then an experiment is designed and conducted that evaluates the software change, finally a decision is made to continue or pivot to another feature. This simplified process is based on the RIGHT model. The roles involved in each phase are shown to the left.

The mapping study [6] by Auer and Felderer investigated the characteristics of the state of research on continuous experimentation. They observed that the intensity of research activities increased from year to year and that there is a high amount of collaboration between industry and academia. In addition, the authors observed that industrial and academic experts contributed equally to the field of continuous experimentation. Concerning the most influential publications (in terms of citations), the authors found that the most common research type among them is experience report. Another observation of the authors was that in total ten different terms were used for the concept of continuous experimentation.

To summarize, the two previous studies discussed continuous experimentation in terms of its applicability in industry sectors, the treatment types and experimentation designs reported, as well as the characteristics of the research in the field. In contrast to these two mapping studies, this study has a far more comprehensive synthesis. Furthermore, the two previous studies improved the rigor and completeness of the search and synthesis procedures.

3. Research method

Based on these two independently published systematic mapping studies [6,7] we conducted a joint systematic literature review. Thus, the presented sets of papers from these two studies were used as starting sets. Forward snowballing was applied, by following the assumption from Wohlin [21] that publications acknowledge previous research. Relevant research publications were identified in the resulting sets. Next, the two sets were merged and the resulting set was studied to answer the respective research questions. Therefore, qualitative narrative syntheses [22] and a thematic synthesis [23] were conducted to answer the research question based on the found literature.

In the following, the research objective and the forward snowballing procedures are presented. Thereafter, the syntheses used to answer the research questions are described. Finally, the threats to validity are discussed.

[2] T-test is often used to compare whether the mean of two groups are equal, based on the t-score $t = \frac{\bar{x}_1 - \bar{x}_2}{s/\sqrt{n}}$, where $\bar{x}$ is mean, s is the standard deviation, and n is the number of data points. The p-value is derived from the t-score through the t-distribution.

[3] A simple approximate power calculation [1] for fixed 95% confidence level and 90% statistical power is $n = (4rs/\Delta)^2$, where n is the number of users, r is the number of groups, s is the standard deviation, and Δ is the effect to detect.

3.1. Research objective

The aim of this research is to give an overview of the current state of knowledge about specific aspects of continuous experimentation. The research questions as stated in the introduction are on: (1) core constituents of a CE framework, (2) technical solutions within CE, (3) challenges with CE, and (4) benefits with CE. Based on the prior mapping studies we observed that there were many papers on models for processes and infrastructure, technical solutions, and challenges for CE and identified these as suitable targets for a systematic review.

3.2. Forward snowballing

The two existing sets of papers emerging from the previous literature reviews [6,7], were used as starting sets for forward snowballing. They were selected as starting sets, because both studies were in the field of continuous experimentation and they had similar research directions. Moreover, both studies were conducted within a short time of each other and had similar inclusion criteria. Hence, the authors are confident that the union of both selected paper sets is a good representation of the field of continuous experimentation in this context until 2017.

The forward snowballing was executed independently for each starting set. After having elaborated a protocol to follow, half of the authors worked on Set A (based on [6], with 82 papers) and half of them on the other Set B (based on [7], with 62 papers). In total, the starting sets contained 100 distinctive papers of which 44 papers were shared among both starting sets. The citations were looked up on Google Scholar.[4] Since the two previous mapping studies covered publications until 2017, the forward snowballing was conducted by considering papers within the time span 2017–2019. The snowballing was executed until no new publications were found.

In the process of snowballing, we used a joint set of inclusion and exclusion criteria. A paper was included if *any* of the inclusion criteria applies, unless *any* of the exclusion criteria applies. The decision was based primarily on the abstract of papers. If this was insufficient to make a decision, the full paper was examined. In doubt, the selection of a paper was discussed with at least one other author. The criteria were defined as such:

Inclusion criteria.

- Any aspect of continuous experimentation (process, infrastructure, technical considerations, etc.)
- Any aspect of controlled experiments (designs, statistics, guidelines, etc.)
- Techniques that complement controlled experiments

Exclusion criteria.

- Not written in English
- Not accessible in full-text
- Not peer reviewed or not a full paper
- Not a research paper: track, tutorial, workshop, talk, keynote, poster, book
- Duplicated study (the latest version is included)
- Primary focus on business-side of experimentation, advertisement, user interface, recommender system

The quality and validity of the included research publications were ensured through the inclusion and exclusion criteria. For instance, publications that did not go through a scientific peer-reviewing process were not considered according to the exclusion criteria. Moreover, to ensure that only mature work was included both vision papers with no evidence based contribution and short papers with preliminary results were excluded.

To summarize, the forward snowballing based on the starting Set A [6] resulted in 100 papers (Set A') and the starting Set B [7] resulted in 88 papers (Set B'). After merging the two paper sets, a total of 128 distinctive papers represent the result of the applied forward snowballing.

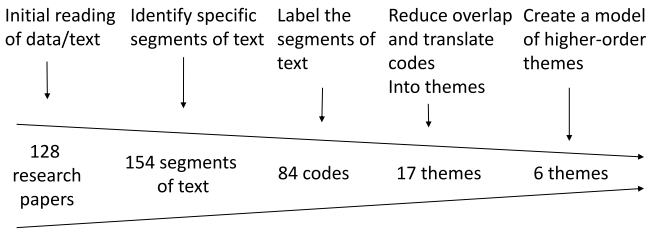

Fig. 2. Applied thematic synthesis process.
Source: Adapted from Cruzes et al. [23,25].

3.3. Synthesis

To answer each research question, the collection of found papers was studied in more detail with respect to the individual research question. Therefore, two qualitative narrative syntheses [22] and one thematic synthesis [23] were conducted.

For the first two research questions a narrative synthesis was conducted for each question. This type of synthesis aggregates qualitative recurring themes within papers and provides a foundation for evidence-based interpretations of the themes in a narrative manner. Thus, the collected set of papers was studied under the heading of the two respective research questions (RQ1, RQ2) to identify relevant themes in it. Next, the found themes were summarized and identified patterns within them were reported. In addition, all papers were classified in terms of their *research type* according to Wieringa et al. [24] to identify what solutions were applied (RQ2). As a result, the findings represent an aggregated view on the components of a continuous experimentation framework (RQ1) and the technical solutions that are applied during experimentation (RQ2). The found components of a continuous experimentation framework are described in Section 4.1. An overview of the identified solutions can be found in Section 4.2.

For the third research question, a thematic synthesis following the steps and checklist proposed by Cruzes and Dybå [23] was conducted (see Fig. 2). In addition, the examples given in Cruzes et al. [25] were consulted. As an initial step, all 128 selected papers were read and in total 154 segments of text were identified. Next, each text segment was labeled with a code. A total of 84 codes were used to characterize the text segments. These codes were loosely based on terms that were identified in previous literature studies [6,7] and evolved during the labeling of the text segments. Thereafter, the codes that had overlapping themes were reduced into 17 themes. In the last step, these 17 themes were arranged according to 6 higher-order themes. The result of this analysis can be found in Section 4.3.

Fig. 3 illustrates the thematic analysis process with the theme "low impact". Based on the reading of five papers, four text segments were extracted. These segments were labeled with the codes benefits, budget and experiment prioritization. In the next step, the common theme among the codes was identified and the codes were reduced to the theme "low impact". During the creation of the model of higher-order themes, this theme was assigned to the higher-order theme "business challenges". All text segments and codes can be found in the results of the study that are available online (see Section 3.4).

[4] https://scholar.google.com/.

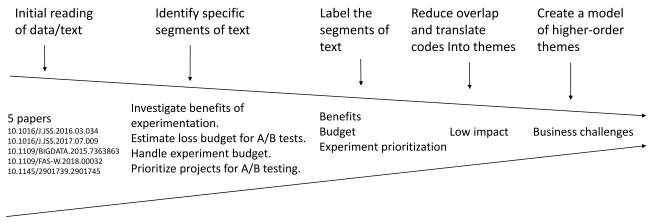

Fig. 3. Schematic example of the thematic synthesis.

3.4. Threats to validity

In every step of this research possible threats to its validity were considered and minimized when possible. In the following, the potential threats are discussed to provide guidance in the interpretation of this work. This section is structured by the four criteria construct validity, internal validity, external validity and reliability by Easterbrook et al. [26].

Construct validity. This threat is about the validity of identification and selection of publications. A challenging threat to overcome is the completeness of the literature search without a biased view of the subject. To mitigate this threat, all papers from the start sets were used without any further exclusion. The larger start set (in comparison to applying the exclusion criteria from the start) was expected to lead to a broader coverage of the literature during the forward snowballing. Furthermore, the process of forward snowballing was adapted in the way that the candidate selection was tolerant about which papers to include, which increases the coverage of the literature search. However, publications may have been falsely excluded because of misjudgment. Nevertheless, we conducted two parallel forward snowballing searches by different authors based on slightly different starting sets, which should mitigate this threat.

Internal validity. Threats that are caused by faulty conclusions could happen because of authors bias at the selection, synthesis of publications and interpretation of the findings. To mitigate this threat, a second author was consulted in case of any doubt. Nevertheless, activities like paper inclusion/exclusion and thematic synthesis inevitably suffer from subjective decisions.

External validity. Threats to external validity covers to which extent the generalization of the results is justified. As the aim of this study is to give an overview of continuous experimentation and to explore the future work items in continuous experimentation, the results should not be generalized beyond continuous experimentation. Therefore, this threat is neglectable.

Reliability. This threat focuses on the reproducibility of the study and the results. To mitigate this threat every step and decision of the study were recorded carefully and the most important decisions are reported. The results of the study are available online [27]. This enables other researchers to validate the decision made on the data. Furthermore, it allows to repeat the study.

4. Results

In this section the results of the literature review are presented according to the research questions.

4.1. What are the core constituents of a CE framework (RQ1)?

To conduct continuous experimentation, an organization has to have some constituents of a framework for experimentation. There is some process involved (implicit or explicit) and some infrastructure is required, which includes a toolchain as well as organizational processes. In the following, both aspects of an experiment, the process and its supporting infrastructure are discussed in detail.

4.1.1. Experiment process

The experiment process can be described in a model that gives a holistic view of the phases and environment around experimentation. Most studies on experiment processes present qualitative models based on interview data. Two models describe the overall process of experimentation. First, the reference model RIGHT (Rapid Iterative value creation Gained through High-frequency Testing) by Fagerholm et al. [38] contains both an infrastructure architecture and a process model for continuous experimentation. The process model builds on the Build, Measure, Learn [28] cycle of Lean Startup. The process in Fig. 1 is a simplified view of RIGHT. Second, the HYPEX (Hypothesis Experiment Data-Driven Development) model is another earlier process model by Holmström Olsson and Bosch [47,160]. In comparison to the RIGHT model, it is less complete in scope, however it does go into further details in hypothesis prioritization using a gap analysis.

Kevic et al. [48] present concrete numbers on the experiment process used at Microsoft Bing through a source code analysis. They have three main findings: (1) code associated with an experiment is larger in terms of files in a changeset, number of lines, and number of contributors; (2) experiments are conducted in a sequence of experiments lasting on average 42 days, where each experiment is on average conducted for one week; and (3) only a third of such sequences are eventually shipped to users.

In addition to the general models described above, several models deal with a specific part of the experiment cycle. The CTP (Customer Touchpoint) model by Sauvola et al. [49] focuses on user collaboration and describes the various ways that user feedback can be involved in the experimentation stages. Amatrian [50] and Gomez-Uribe and Hunt [34] describe their process for experimentation on their recommendation system at Netflix, in particular how they use offline simulation studies with online controlled experiments. In the ExG Model (Experimentation Growth), by Fabijan et al. [51,52], organizations can quantitatively gauge their experimentation on technical, organizational, and business aspects. In another model by Fabijan et al. [53] they describe the process of analyzing the results of experiments and present a tool that can make the process more effective, by e.g. segmenting the participants automatically and highlighting the presence of outliers. Finally, Mattos et al. [54] present a model that discuss details on activities and metrics on experiments.

4.1.2. Infrastructure

Depending on what type of experimentation is conducted, different infrastructure is required. For controlled experimentation, in particular, technical infrastructure in the form of an *experimentation platform* is critical to increase the scale of experimentation. At the bare minimum it needs to divide users into experiment groups and report statistics. Gupta et al. [39] at Microsoft have detailed the additional functionality of their experimentation platform. Also Schermann et al. [55] have described attributes of system and software architecture suitable for experimentation, namely, that micro-service-based architectures seem to be favored. Some experimentation platforms are specialized to specific needs: automation [56,158], or describing deployment through formal models [57], or how experimentation can be supported by non-software engineers [58,59].

There are also non-technical infrastructure requirements, regardless of the type of experimentation in use. The required roles are [43]: data scientists, release engineers, user researchers, and the standard software engineering roles. Also, an organizational culture [37,60] that is open towards experimentation is needed. For example, Kohavi et al. [37] explain that managers can hinder experimentation if they overrule results with their opinions. They call the phenomenon the highest paid persons opinion (HiPPO).

While experimentation is typically associated with large companies, like Microsoft or Facebook, there are three interview studies that discuss experimentation at startups specifically [38,61,62]. As argued by Gutbrod et al. [62], startup companies often guess or estimate roughly

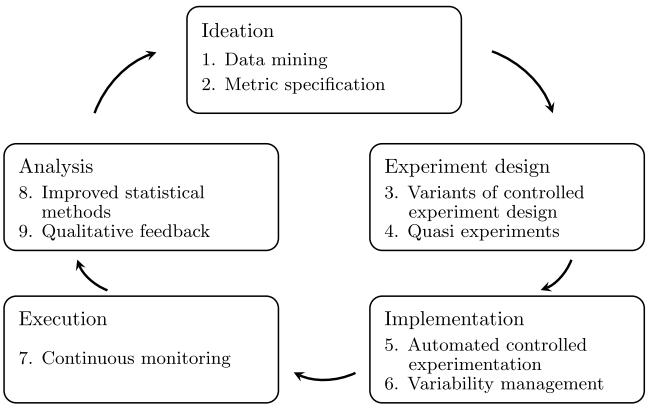

Fig. 4. Solutions applied within continuous experimentation arranged by main phase of experimentation from Section 2.2.

about the problems and customers they are addressing. Thus, there is a need for startup companies to be more involved with experimentation, although they have less infrastructure in place.

Finally, we would like to call attention to some of the few case studies and experience reports on experimentation on "ordinary" software companies, which are neither multi-national corporations nor startups [46,63,64]; in e-commerce, customer relations, and gaming industry respectively. None of these papers are focused on infrastructure, but do mention that infrastructure needs to be implemented. Risannen et al. [63] mentions additional challenges when infrastructure must be implemented on top of a mature software product. In summary, this indicates that infrastructure requirements are modest unless scaling up to multi-national corporation levels with millions of users.

4.2. What technical solutions are applied in what phase within CE (RQ2)

The study of the selected publications revealed many different types of solutions that were summarized by common themes. Fig. 4 gives an overview of the identified solutions organized in the phases of experimentation in Fig. 1.

4.2.1. Data mining

Data from previous experiments can be used to make predictions or mine insights to either improve the reliability of the experiment or for ideation. There were three specific solutions for data mining in continuous experimentation: (1) calculating variance of metrics through bigger data sets than just one experiment at Netflix [65], Microsoft [66,67], Google [68], and Oath [69]; (2) mining for invalid tests through automatic diagnosis rules at LinkedIn [70] and Sevenval Technologies [71]; and finally (3) to extract insights from segments of the users, by detecting if a treatment is more suitable for those specific circumstances [72], this technique is applied at Snap [73] and Microsoft [53].

4.2.2. Metric specification

Defining measurements for software is difficult. At Microsoft they have hundreds of metrics in place for each experiment, they recommend organizing metrics in a hierarchy [40] and evaluating how well metrics work [74,75]. At Yandex, they pair OEC metrics with a statistical significance test to create an overall acceptance criteria (OAC) instead [76]. Several pieces of work are on defining and improving usability metrics, especially from Yandex [77–79]. Also at Microsoft they have a rule-based classifier where each user action is either a frustration or benefit signal [80].

Some general guidelines for defining metrics follow. At Microsoft [40,74,75], they have hundreds of metrics for each experiment (in addition to a few OEC). Machmouchi and Buscher [40] from Microsoft describe how their metrics are interpreted in a hierarchy in their tool (similar to Fabijan et al. [53] also at Microsoft). At the top of the hierarchy are statistically robust metrics (meaning they tend not to give false positives) and at the bottom are feature specific metrics that are allowed to be more sensitive. They have also developed methods to evaluate how well metrics work. Dmitriev et al. [75] give an experience report on how metrics are evaluated at Microsoft system in practice. Deng et al. [74] define metrics for evaluating metrics: directionality and sensitivity. They measure respectively whether a change in the metric aligns with good user experience and how often it detects a change in user experience.

Usability metrics are hard to define since they are not directly measurable without specialized equipment, such as eye-tracking hardware or focus groups. The measurements that are available, such as clicks or time spent on the site, do not directly inform on whether a change is an improvement or degradation in user experience. In addition, good user experience does not necessarily correlate positively with business value, e.g. clickbait titles for news articles are bad user experience but generate short term revenue. Researchers from Yandex [77–79,81–83] are active in this area, with the following methods focused on usability metrics: detecting whether a change in a metric is a positive or negative user experience [81]; learning sensitive combinations of metrics [79]; quantifying and detecting trends in user learning [78]; predicting future behavior to improve sensitivity [82]; applying machine learning for variance reduction [83]; and finally correcting misspecified usability metrics [77]. Machmouchi et al. [80], at Microsoft, designed a rule-based classifier where each user action is either a frustration or benefit signal; the tool then aggregates all such user actions taken during a session into a single robust utility metric.

4.2.3. Variants of controlled experiments design

Most documented experiments conducted in industry are univariate A/B/n-tests [7], where one or more treatments are tested against a control. Extensions to classical designs include a two-staged approach to A/B/n tests [84] and a design to estimate causal effects between variables in a multivariate test (MVT) [85]. MVTs are cautioned against [86] because of their added complexity. In contrast, other researchers take an optimization approach using lots (see Section 4.2.5) of variables with multi-armed bandits [46,87–89] or search-based methods [90–92]. Also mixed methods research is used to combine quantitative and qualitative data. Controlled experiments require deployment, feedback from users at earlier stages of development can thus be cheaper. There are works on combining results of such qualitative methods [93] and collecting it in parallel with A/B tests [94].

4.2.4. Quasi-experiments

A quasi-experiment (or natural experiment) is an experiment that is done sequentially instead of in parallel; this definition is the same as in empirical research in general [20]. The reason for doing it is that it has a lower technical complexity. In fact, any software deployment can have its impact measured by observing the effect before and after deployment. The drawback of this is that analyzing the results can be difficult due to the high risk of having external changes affect the result. That is, if anything extraordinary happens roughly at the same time as the release it might not be possible to properly isolate the results. Since the world of software is in constant change the use of quasi-experiments is challenging. The research directions on quasi-experiments involve how to eliminate external sources of noise to get more reliable results. This is studied at Amazon [95] and Linkedin [96], particularly for environments were control is continuous deployment is hard (such as mobile app development).

4.2.5. Automated controlled experimentation with optimization algorithms

With an optimization approach, the allocation of users to the treatment groups is dynamically varied to optimize an OEC, such that treatments that perform well continuously receive more and more traffic over time. With sufficient automation, these techniques can be applied to lots of treatment variables simultaneously. This is not a replacement for classical designs; in an interview study by Ros and Bjarnason [46], they explain that such techniques are often validated themselves using A/B tests. In addition, based on the studies included here, only certain parameters are eligible, such as the design and layout of components in a GUI, or parameters to machine learning algorithms or recommender systems. Some of these optimizations are black-box methods, where multiple variables are changed simultaneously and with little opportunity to make statistical inferences from the experiments.

Tamburelli and Margara [92] proposed search-based methods (i.e. genetic algorithms) for optimization of software, and Iitsuka and Matsuo [97] demonstrated a local search method with a proof of concept on web sites. Miikkulainen [90], at Sentient Technologies, have a commercial genetic algorithm profiled for optimizing e-commerce web sites. Bandit optimization algorithms are also used in industry at Amazon [88] and AB Tasty [87], it is a more rigorous formalism that requires the specification of a statistical model on how the OEC behaves. Ros et al. [91] suggested a unified approach of genetic algorithms and bandit optimization. Similar algorithms exist to handle continuous variables, as is needed for hardware parameters [89,98] and for optimizing machine learning and compiler parameters [45].

Two studies apply optimization [99,100] to scheduling multiple standard A/B tests to users, where only a single treatment is administered to each user. The idea is to optimize an OEC without sacrificing statistical inference.

4.2.6. Variability management

Experimentation 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. In terms of an experiment platform, this can be part of the *experiment execution service* and/or the *experimentation portal* [39].

There have been attempts at imposing systematic constraints and structure in the configuration of how the variables under experimentation interact with formal methods. Cámara and Kobsa [101] suggest using a feature model of the software parameters in all experiments. This work has not advanced beyond a proof-of-concept stage.

Neither in our study, nor in the survey by Schermann et al. [55], is there any evidence of formal methods in a dynamic and constantly changing experimentation environment. The focus of the tools in actual use are rather on flexibility and robustness [102,103]. Rahman et al. [104] studied how feature toggles are used in industry. Feature toggles are ways of enabling and disabling features after deployment, as such they can be used to implement A/B testing. They were found to be efficient and easy to manage but adds technical debt.

A middle ground between formal methods and total flexibility has evolved in the tools employed in practice. Google has proprietary tools in place to manage overlapping experiments in large scale [103]. In their tools, each experiment can claim resources used during experimentation and a scheduler ensures that experiments can run in parallel without interference. Facebook has published an open-source framework (PlanOut) specialized for configuring and managing experiments [102], it features a namespace management system for experiments running iteratively and in parallel. SAP has a domain-specific language [105] for configuring experiments that aims at increasing automation. Finally, Microsoft has the ExP platform, but none of the selected papers focus solely on the variability management aspect of it.

4.2.7. Improved statistical methods

The challenges with experimentation motivate improved statistical techniques specialized for A/B testing. There are many techniques for fixing specific biases, sources of noise, etc: a specialized test for count data at SweetIM [36]; fixing errors with dependent data at Facebook [106]; improvements from the capabilities of A/A testing on diagnosis (which tests control vs control expecting no effect) at Yahoo [107] and Oath [108]; better calculation of overall effect for features with low coverage at Microsoft [109]; fixing errors from personalization interference at Yahoo [110]; fixing tests under telemetry loss at Microsoft [111]; correcting for selection bias at Airbnb [112]; and algorithms for improved gradual ramp-up at Google [113] and LinkedIn [114].

4.2.8. Continuous monitoring

Aborting controlled experiments pre-maturely in case of outstanding or poor results is a hotly debated topic on the internet and in academia, under the name of continuous monitoring, early stopping, or continuous testing. The reason for wanting to stop early is to reduce opportunity costs and to increase development speed. It is studied by Microsoft [115], Yandex [116], Optimizely [117], Walmart [118], and Etsy [119]. This concept is similar to the continuous monitoring used by researchers in the DevOps community and continuous software engineering [43] where it refers to the practice of monitoring a software system and sending alerts in case of faults. The issue with continuous monitoring of experiments is the increased chance of getting wrong results if carried out incorrectly. Traditionally, the sample size of an experiment is defined beforehand through a power calculation. If the experiment is continuously monitored with no adjustment, then the results will be skewed with inflated false negative and positive error rates.

4.2.9. Qualitative feedback

While the search strategy in this work was focused on controlled experiments, research on qualitative feedback was also included from experience reports on using many different types of feedback collecting methods, for example at Intuit [93,120] and Facebook [33]. The qualitative methods are used as complements to quantitative methods, either as a way to better explain results or as a way to obtain feedback earlier in the process, before a full implementation is built. That is, qualitative feedback can be collected on early user experience sketches or mock-ups. Another use of qualitative methods is to elicit hypotheses that can be used as a starting point for an experiment. Examples of methods include focus groups, interviews, and user observations.

In addition, at Unister [94] the authors explain how they collect qualitative user feedback in parallel with A/B tests, such that the feedback is split by experiment group. According to the authors, this seems to be a way to get the best of both quantitative and qualitative worlds. It does require implementing a user interface for collecting the feedback in a non-intrusive way in the product. Also, the qualitative feedback will not be of as high quality as when it is done in person with e.g. user observation or focus groups.

4.3. What are the challenges with continuous experimentation (RQ3)?

Continuous experimentation encompasses a lot of the software engineering process, it requires both infrastructure support and a rigorous experimentation process that connects the software product with business value. As such, many things can go wrong and the challenges presented here is an attempt at describing such instances. Most of the research on challenges is evaluation research, with interviews or experience reports. Many of the challenges are severe, in that they present a hurdle that must be overcome to conduct continuous experimentation. A failure in any of the respective category of challenges will make an experiment: unfeasible due to technical reasons, not considered by unresponsive management, untrustworthy due to faulty use of statistics, or without a business case. The analysis of the papers revealed six categories of challenges (see Table 1) that are discussed in the following in more detail.

Table 1
Summary of challenges with continuous experimentation per category with description and key references that focus on them.

Challenge	Description	References
1. Cultural, organizational, and managerial challenges		
Knowledge building	There are many roles and skills required, so staff need continuous training.	[35,63,121]
Micromanagement	Experimentation requires management to focus on the process (c.f. HiPPO in Section 4.3.1).	[37]
Lack of adaption	Engineers need to be onboarded on the process as well as managers.	[44]
Lack of communication	Departments and teams should share their results to aid each other.	[63,121]
2. Business challenges		
Low impact	Experimentation might focus efforts on incremental development with insufficient impact.	[43,122]
Relevant metrics	The business model of a company might not facilitate easy measurement.	[44,52,123]
Data leakage	Companies expose internal details about their product development with experimentation.	[124]
3. Technical challenges		
Continuous delivery	The CI/CD pipeline should be efficient to obtain feedback fast.	[44,52,55]
Continuous deployment	Obstacles exists to putting deliveries in production, e.g. on-premise installations in B2B.	[63]
Experimental control	Dividing users into experimental groups have many subtle failure possibilities.	[42,125,126]
4. Statistical challenges		
Exogenous effects	Changes in environment can impact experiment results, e.g. trend effects in fashion.	[126,127]
Endogenous effects	Experimentation itself causes effects, such as carry-over or novelty effects.	[126,128]
5. Ethical challenges		
Data privacy	GDPR gives users extensive rights to their data which companies must comply with.	[129]
Dark patterns	A narrow focus on numbers only can lead to misleading user interfaces.	[130]
6. Domain specific challenges		
Mobile	The app marketplaces impose constraints on deployment and variability.	[64,96,131]
Cyber–physical systems	Making continuous deployments can be infeasible for cyber–physical systems.	[132–134]
Social media	Users of social media influence each other which impacts the validity of experiments.	[135–137]
E-commerce	Experimentation needs to be able to differentiate effects from products and software changes.	[138,139]

4.3.1. Cultural, organizational, and managerial challenges

The challenges to organizations and management are broad in scope, including: difficulty in changing the organizational culture to embrace experimentation [44]; building experimentation skills among employees across the whole organization [35,121]; and finally communicating results and coordinating experiments in business to business, where there are stakeholders involved across multiple organizations [63,121].

A fundamental challenge that has to be faced by organizations adopting continuous experimentation, is the shift from the highest-paid person's opinion (HiPPO) [1,37] to data-driven decision making. If managers are used to making decisions about the product then they might not take account of experimental results that might run counter to their intuition. Thus, decision-makers must be open to have their opinions changed by data, else the whole endeavor with experimentation is useless.

4.3.2. Business challenges

The premise behind continuous experimentation is to increase the business value of software development efforts. The most frequent challenge in realizing this is defining relevant metrics that measure business value [44,52,121,123,126]. In some instances the metric is only indirectly connected to business, for example in a business-to-business (B2B) company with a revenue model that is not affected by the software product, then improving product quality and user experience will not have a direct business impact. Also, the impact of experiments might not be sufficient in terms of actual effect [86,122]. Fitzgerald and Stol [43] argue that continuous experimentation and innovation can lead to incremental improvements only, at the expense of more innovative changes that could have had a bigger impact. Another business challenge of continuous experimentation was highlighted by Conti et al. [124]; they crawled web sites repeatedly and tried to automatically detect a difference in server responses. Thereby they showed how easily such data leakage can facilitate industrial espionage on what competitors are developing.

4.3.3. Technical challenges

Efficient continuous deployment facilitates efficient experimentation. Faster deployment speed shortens the delay between a hypothesis and the result of an experiment. The ability to have an efficient continuous delivery cycle is cited as a challenge both for large [42] and small companies [44,52,55]. In addition, continuous deployment is further complicated in companies involved in business to business (B2B) [63], where deployment has multiple stakeholders involved over multiple organizations.

In a laboratory experiment setting, it is possible to control variables such as ensuring homogeneous computer equipment for all groups and ensuring that all groups have equal distribution in terms of gender, age, education, etc. For online experiments, such controls are much harder due to subtle technical reasons. Examples therefore are: users assigned incorrectly to groups due to various bugs [42]; users changing groups because they delete their browsing history or multiple persons share the same computer [42,126,140,141]; and robots from search engines cause abnormal traffic affecting the results [125,142].

4.3.4. Statistical challenges

Classical experimental design as advocated by the early work on continuous experimentation and A/B-testing [1] does not account for time series. Not only can it be hard to detect the presence of effects related to trends, but they can also have an effect on the results. Some of these trend effects occur due to outside influence, so-called *exogenous effects*, for example, due to seasonality caused by fashion or other events which can affect traffic [126,127]. With domain knowledge, these effects can be accounted for [157]. For example in e-commerce, experiment results obtained during Christmas shopping week might not transfer to the following weeks.

Other statistical challenges are caused by the experimentation itself, called *endogenous effects*, such as the carryover effect [127,128] where the result of an experiment can affect the result of a following experiment. There are also endogenous effects caused intentionally, through what is known as ramp up, where the traffic to the test group is initially low (such as 5%/95%) and incrementally increased to the full 50%/50% split. This is done to minimize the opportunity cost of a faulty experiment design. It can be difficult to analyze the results of such experiments [125,142]. Furthermore, learning and novelty effects where the users change their impression of the feature after using it for a while are challenging [126,128].

Endogenous effects will be hard to foresee until experimentation is implemented in a company. As such, handling statistical challenges is an ongoing process that will require more and more attention as experimentation is scaled up.

4.3.5. Ethical challenges

Whenever user data is involved there is a potential for ethical dilemmas. When Yaman et al. [129] surveyed software engineering practitioners, the only question they agreed on was that users should be notified if personal information is collected. Since GDPR went into effect in 2018 this is now a requirement. Jian et al. [130] investigate how A/B testing tools are used in illegal discrimination for certain demographics, e.g., by adjusting prices or filtering job ads. These are examples of what is known as *dark patterns* in the user experience (UX) research community [29]. The study was limited to sites using front end Optimizely (a commercial experimentation platform) metadata.

4.3.6. Domain specific challenges

Some software sectors have domain-specific challenges or techniques required for experimentation, of which in the analysis of the papers four prominent domains were found: (1) mobile apps, (2) cyber–physical systems, (3) social media, and (4) e-commerce. Whether or not all of these concerns are domain-specific or not is debatable. However, these studies were all clear on what domain their challenges occurred in.

There is a bottleneck in continuous deployment to the proprietary application servers of Android Play or Apple's App Store, which imposes a bottleneck on experimentation for mobile apps. Lettner et al. [131] and Adinata and Liem [143] have developed libraries that load new user interfaces at run time, which would otherwise (at the time of writing in year 2013 and 2014 respectively) require a new deployment on Android Play. Xu et al. [96] at LinkedIn instead advocate the use of quasi-experimental designs. Finally, Yaman et al. [64] have done an interview study on continuous experimentation, where they emphasize user feedback in the earlier stages of development (that do not require deployment).

Embedded systems, cyber–physical systems, and smart systems face similar challenges to mobile apps, namely continuous deployment. None of the studied publications of this study claims widespread adoption of experimentation at an organizational level. This suggests that research of experimentation for embedded software is in an early stage. Mattos et al. [134] and Bosch and Holmström Olsson [132] outline challenges and research opportunities in this domain, among them are: continuous deployment, metric definition, and privacy concerns. Bosch and Eklund [41,144] describe required architecture for experimentation in this domain with a proof-of-concept on vehicle entertainment systems. Giaimo et al. [133,145] cite safety concerns and resource constraints for the lack of continuous experimentation.

The cyber–physical systems domain also includes experimentation where the source of noise is not human users, but rather hardware. The research on self-adaptive systems overlap with continuous experimentation: Gerostathopoulos et al. [146] have described an architecture for how self-adaptive systems can perform experimentation, with optimization algorithms [147] that can handle non-linear interactions between hardware parameters [148]. In addition, two pieces of work [149,150] on distributed systems focus on experimentation, with a survey and a tool on how distributed computing can support experimentation for e.g. cloud providers.

Backstrom et al. [136] from Facebook describe that users of social media influence each other across experiment groups (thus violating the independence assumption of statistical tests); they call it the network effect. It is also present at Yahoo [151] and LinkedIn [60,152,153]. The research on the network effect includes: ways of detecting it [153], estimating its effect on cliques in the graph [135,137], and reducing the interference caused from it [154].

The final domain considerations come from e-commerce. At Walmart, Goswami et al. [138] describe the challenges caused by seasonality effects during holidays and how they strive to minimize the opportunity cost caused by experimentation. At Ebay, according to Wang et al. [139], the challenges are caused by the large number of auctions that they need to group with machine learning techniques for the purpose of experimental control.

4.4. What are the benefits with continuous experimentation (RQ4)?

Many authors mention the benefits of CE only in passing as motivation [37,120], few papers explicitly mention them (e.g. [155]).

Bosch [120] mentions the reduced cost of collecting passive customer feedback with continuous experimentation in comparison with active techniques like surveys. Also, Bosch claims that customers have come to expect software services to continuously improve themselves and that experimentation can provide the means to do that in a process that can be visible to users. Kohavi et al. [37] claim that edge cases that are only relevant for a small subset of users can take a disproportionate amount of the development time. Experimentation is argued for as a way to focus development, by first ensuring that a feature solves a real need with a small experiment and then optimizing the respective feature for the edge cases with iterative improvement experiments. In this way, unnecessary development on edge cases can be avoided if a feature is discarded early on.

Fabijan et al. [155] focus solely on benefits, differentiated between three levels as follows. (1) In the *portfolio level*, the impact of changes on the customer as well as business value can be measured which is of great benefit to company-wide product portfolio development. (2) In the *product level*, the product receives incrementally improvement quality and reduced complexity by removing unnecessary features. Finally, (3) in the *team level of benefits*, the findings of experiments support the related teams to prioritize their development activities given the lessons learned from the conducted experiments. Another benefit for teams with continuous experimentation is that team goals can be expressed in terms of metric changes and their progress is measurable.

5. Discussion

This study builds on two prior independent mapping studies to provide an overview of the conducted research. This review has been conducted to answer four research questions that can guide practitioners. In the following, the results of the study are discussed for each research question, in the form of recommendations to practitioners and implications for researchers.

5.1. Required frameworks (RQ1)

The first research question (RQ1) about the core constituents of a framework for continuous experimentation revealed two integral parts of experimentation, the *experimentation process* and the technical as well as organizational *infrastructure*.

5.1.1. Process for continuous experimentation

In the literature, several experimentation process models were found on the phases of conducting online controlled experimentation. They describe the overall process [38], represent the established experiment process of organizations [48], or cover specific parts of the experiment cycle [53]. Given that all models describe a process with the same overall objective of experimentation, it can become difficult to decide between them. Two reference models are published [38,47], which may be used as a basis for future standardization of the field. Future research is needed to give guidance in the selection between models and variants.

Many of the experience reports [86,142] warn about making experiments with too broad scope, instead they recommend that all experiments should be done on a minimum viable product or feature [38]. However, the warnings all come from real lessons learned caused by having done such expensive experiments. We believe that the current process models do not put sufficient emphasis on conducting small experiments. For example, they could make a distinction between prototype experiments and controlled experiments on a completed project. That way if the prototype reveals flaws in the design it avoids a full implementation.

As such, our recommendation to practitioners in regards to process is to follow one of the reference experimentation processes [38,47] and in addition add the following two steps to minimize the cost of experiments. First, to spend more time before experimentation to ensure that experiments are really on a minimum viable feature by being diligent about what requirements are strictly needed at the time. Second, that experiments should be pre-validated with prototypes, data analysis, etc.

5.1.2. Infrastructure for continuous experimentation

The research on the infrastructure required to enable continuous experimentation was primarily focused on large scale applications within mature organizations (e.g. Microsoft [39]). One reason for this focus may be the large number of publications (e.g. experience reports) from researchers associated with large organizations. The large number of industrial authors indicates a high interest of practitioners in the topic. However, it should not restrict the community's focus on large scale applications only. The application of continuous experimentation within smaller organizations has many open research questions. These organizations provide additional challenges on experimentation because of their probably less already existing infrastructure and smaller user base. For example, the development of sophisticated experimentation platforms may not be feasible in the extent to which it is for large organizations. Thus, lightweight approaches to experimentation that do not require large up-front investments could make experimentation more accessible to smaller organizations.

Technical infrastructure has not been reported as being a significant hurdle for any of the organizations in which continuous experimentation was introduced in this study. The technical challenges seem to appear later on when the continuous experimentation process has matured and the scale of experimentation needs to ramp up. Rather, the organizational infrastructure seems to be what might cause an inability to conduct experimentation. The challenges presented in Section 4.3 support this claim too, so the more severe infrastructural requirements appear to be organizational [43] and culture oriented [37,60], at least to get started with experimentation. The reason for this is that experimentation often involves decision making that traditionally fall outside the software development organization. For example, deciding on what metric software should be optimized for might even need to involve the company board of directors. Following that, the recommendation to practitioners is to not treat continuous experimentation as a project that can be solved with only software development. The whole organization needs to be involved, e.g., to find metrics and ensuring that the user data to measure this can be acquired. Otherwise, if the software development organization conducts experimentation in isolation, the soft aspects of infrastructure might be lacking or the software might be optimized with the wrong goal in mind.

5.2. Solutions applied (RQ2)

Concerning the solutions that are applied within continuous experimentation (RQ2), the literature analysis revealed solutions about qualitative feedback, variants of controlled experiments design, quasi-experiments, automated controlled experimentation with optimization algorithms, statistical methods, continuous monitoring, data mining, variability management, and metric specification. For each of these solutions in literature, themes were proposed. One observation made was that the validation of most proposed solutions could be further improved by providing the used data sets, a context description or the necessary steps that allow to reproduce the presented results. Also, many interesting solutions would benefit from further applications that demonstrate their applicability in practice. Another observation was that many solutions are driven by practical problems of the author's associated organization (e.g. evaluation of mobile apps [95]). This has the advantage that the problems are of relevance for practice and the provided solutions are assumed to be applicable in similar contexts. Publications of this kind are guidelines for practitioners and valuable research contributions.

There are a lot of solutions for practitioners to choose from, most of them solve a very specific problem that has been observed at a company. In Fig. 4, the solutions are arranged by phase of the experimentation process. What follows is additional help to practitioners to know what solution to apply for a given problem encountered, which is in the design science tradition known as technological rules [30]:

- to achieve *additional insights* in concluded experiments apply *(1) data mining* that automatically segments results for users' context;
- to achieve *more relevant results* in difficult to measure software systems apply *(2) metric specification techniques*.
- to achieve *richer experiment feedback* in continuous experimentation apply *(3) variants of controlled experiments design* or *(9) qualitative feedback*.
- to achieve *quantitative results* in environments where parallel deployment is challenging apply *(4) quasi-experiments*;
- to achieve *optimized user interfaces* in software systems that can be evaluated on a single metric apply *(5) automated controlled experimentation with optimization algorithms*;
- to achieve *higher throughput of experiments* in experimentation platforms apply *(6) variability management techniques* to specify overlapping experiments;
- to achieve *trustworthy results* in online controlled experiments apply *(7) improved statistical methods* or *(1) data mining* to calibrate the statistical tests;
- to achieve *faster results* in online controlled experiments apply *(8) continuous monitoring* to help decide when experiments can be stopped early.

5.3. Challenges (RQ3)

Many authors of the studied literature mentioned challenges with continuous experimentation in their papers. The thematic analysis of the challenges identified six fundamental challenge themes. Here they are presented along with the recommendations to mitigate the risks.

The *cultural, organizational and managerial* challenges seem to indicate that the multi-disciplinary characteristic of continuous experimentation introduces new requirements to the team. It requires amongst others the collaboration of cross-functional stakeholders (i.e. business, design, and engineering). This can represent a fundamental cultural change within an organization. Hence, the adaption of continuous experimentation involves technical as well as cultural changes. Challenges like the lack of adaption support this interpretation. Mitigating these challenges involves taking a whole organizational approach to continuous experimentation so that both engineers and managers are in agreement about conducting experimentation.

Another theme among challenges is *business*. The challenges assigned to this theme highlight that continuous experimentation has challenges in its economic application with respect to the financial return on investment. The focus of experimentation needs to be managed appropriately in order to prevent investing in incremental development with insufficient impact. Also, that changes cannot be measured with a relevant metric is another business challenge. One possible approach for further research on these challenges could be the transfer from solutions in other disciplines to continuous experimentation. An example therefore is the overall evaluation criteria [31] that was adapted to continuous experimentation by Kohavi et al. [1]. As with the previous challenge theme, this theme of challenges does not have an easy fix. It might be the case that experimentation is simply not applicable for all software companies but further research is needed to determine this.

Concerning the *technical* challenges, the literature review showed that there are challenges related to continuous deployment/delivery and experiment control. The delivery of changes to production is challenging especially for environments that are used to none or infrequent updates, like embedded devices. For such edge cases, new deployment strategies have to be found that are suitable for continuous experimentation. Although solutions from continuous deployment seem to be fitting, they need to be extended with mechanisms to control the experiment at run-time (e.g. to stop an experiment). This can be challenging in environments for which frequent updates are difficult. There is proof-of-concept research [41] to handle these challenges so they do not seem to be impossible blockers to get started on experimentation.

The *statistical* challenges mentioned in the studied literature indicate that there is a need for solutions to cope with the various ways that the statistical assumptions done in a controlled experiment are broken by changes in the real world. There are both changes in the environment (exogenous) and changes caused by experimentation (endogenous). Changes in the environment (e.g. the effect of an advertisement campaign run by the marketing department) can alter the initial situation of an experiment and thus may lead to wrong conclusions about the results. Therefore, the knowledge about an experiment's environment and possible influences needs to be systematically advanced and the experiments themselves should be designed to be more robust. Mitigating these challenges involves identifying and applying the correct solution for the specific problem. There is further research opportunity to document and synthesize such problem-solution pairs.

Ethical aspects are not investigated by many studies. The experience reports and lessons learned publications do not, for example, mention user consent or user's awareness of participation. Furthermore, ethical considerations about which experiments should be conducted or not were seldom discussed in the papers. There were still two challenges identified in this study, involving data privacy and dark patterns. However, examples like the Facebook emotional manipulation study, which changed the user's news feed to determine whether it affects the subsequent posts of a user, show the need for ethical considerations in experimentation [32]. Although this was an experiment in the context of an academic study in psychology, the case nevertheless shows that there are open challenges on the topic of ethics and continuous experimentation. There is not enough research conducted for a concrete recommendation other than raising awareness of the existence of ethical dilemmas involving experimentation.

Continuous experimentation is applied in various domains that require *domain specific* solutions. The challenges on continuous experimentation range from infrastructure challenges, over measurement challenges, to social challenges. Examples are the challenge to deploy changes in cyber–physical systems (infrastructural challenge), to differentiate the effects of one change from another (measurement challenge), and the influence of users on each other across treatment groups (social challenge). Each challenge is probably only relevant for certain domains, however the developed solutions may be adaptable to other domains. Thus, the research on domain-specific challenges could take optimized solutions for specific domains to solutions for other domains.

5.4. Benefits (RQ4)

In many publications about continuous experimentation the benefits of experimentation are mentioned as motivation only; i.e. it increases the quality of the product based on the chosen metrics. The two publications on explicit benefits [120,155] mention improvements not only on the product in business-related metrics and usability but also on the product portfolio offering and generic benefits for the whole organization (better collaboration, prioritization, etc.). More studies are needed to determine, e.g., if there are more benefits, whether the benefits apply for all companies involved with experimentation, or whether the benefits could be obtained through other means. Another benefit is the potential usage of continuous experimentation for software quality assurance. Continuous experimentation could support or even change the way quality assurance is done for software. Software change, for example, could only be deployed if key metrics are not degraded in the related change experiment. Thus, quality degradation could become quantifiable and measurable. Although some papers, like [155], mention the usage of continuous experimentation for software quality assurance.

6. Conclusions

This paper presents a systematic literature review of the current state of controlled experiments in continuous experimentation. Forward snowballing was applied on the selected paper sets of two previous mapping studies in the field. The 128 papers that were finally selected, were qualitatively analyzed using thematic analysis.

The study found two constituents of a continuous experimentation framework (RQ1): an experimentation process and a supportive infrastructure. Based on experience reports that discuss failed experiments in the context of large-scale software development, the recommendation to practitioners is to apply one of the published processes, but also expand it by placing more emphasis on the ideation phase by making prototypes. As for the infrastructure, several studies discuss requirements for controlled experiments to ramp up the scale and speed of experimentation. Our recommendation for infrastructure is to consider the organizational aspects to ensure that, e.g., the necessary channels for communicating results are in place.

Ten themes of solutions (RQ2) were found that were applied in the various phases of controlled experimentation: data mining, metric specification, variants of controlled experiment design, quasi-experiments, automated controlled experimentation, variability management, continuous monitoring, improved statistical methods, and qualitative feedback. We have provided recommendations on what problem each solution theme solves for what context in the discussion.

Finally, the analysis of challenges (RQ3) and benefits (RQ4) of continuous experimentation revealed that only two papers focused explicitly on the benefits of experimentation. In contrast, multiple papers focused on challenges. The analysis identified six themes of challenges: cultural/organizational, business, technical, statistical, ethical, and domain-specific challenges. While the papers on challenges do outnumber the papers on benefits, there is no cause for concern, as the benefits to product quality are also mentioned in many papers as motivation to conduct the research. The challenges to experimentation also come with recommendations in the discussion on how to mitigate them.

As a final remark, we encourage practitioners to investigate the large body of highly industry-relevant research that exists for controlled experimentation in continuous experimentation and for researchers to follow the many remaining gaps in literature revealed within.

CRediT authorship contribution statement

Florian Auer: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Resources, Data curation, Writing - original draft, Writing - review & editing, Visualization. **Rasmus Ros:** Conceptualization, Methodology, Validation, Formal analysis, Investigation, Resources, Data curation, Writing - original draft, Writing - review & editing, Visualization. **Lukas Kaltenbrunner:** Resources, Investigation. **Per Runeson:** Supervision, Writing - review & editing. **Michael Felderer:** Supervision, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partially supported by the Wallenberg Artificial Intelligence, Autonomous Systems and Software Program (WASP) funded by Knut and Alice Wallenberg Foundation and the Austrian Science Fund (FWF): I 4701-N.

References

[1] R. Kohavi, R.M. Henne, D. Sommerfield, 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, ACM, 2007, pp. 959–967, https://doi.org/10.1145/1281192.1281295.

[2] N.E. Fenton, M. Neil, A critique of software defect prediction models, IEEE Trans. Softw. Eng. 25 (5) (1999) 675–689.

[3] R.S. Wahono, A systematic literature review of software defect prediction, J. Softw. Eng. 1 (1) (2015) 1–16.

[4] E. Ronchieri, M. Canaparo, Metrics for software reliability: a systematic mapping study, J. Integr. Des. Process Sci. 22 (2) (2018) 5–25.

[5] D. McKinley, Design for continuous experimentation: Talk and slides, 2012, Online, retrieved August 2019. URL http://mcfunley.com/design-for-continuous-experimentation.

[6] F. Auer, M. Felderer, Current state of research on continuous experimentation: A systematic mapping study, in: 2018 44th Euromicro Conference on Software Engineering and Advanced Applications, SEAA, IEEE, 2018, pp. 335–344, https://doi.org/10.1109/SEAA.2018.00062.

[7] R. Ros, P. Runeson, Continuous experimentation and A/B testing: A mapping study, in: Proceedings of the 4th International Workshop on Rapid Continuous Software Engineering, RCoSE, ACM, 2018, pp. 35–41, https://doi.org/10.1145/3194760.3194766.

[8] M. Shahin, M.A. Babar, L. Zhu, Continuous integration, delivery and deployment: A systematic review on approaches, tools, challenges and practices, IEEE Access 5 (2017) 3909–3943, https://doi.org/10.1109/ACCESS.2017.2685629.

[9] C. Ebert, G. Gallardo, J. Hernantes, N. Serrano, DevOps, IEEE Softw. 33 (3) (2016) 94–100, https://doi.org/10.1109/MS.2016.68.

[10] A. Fabijan, H.H. Olsson, J. Bosch, Customer feedback and data collection techniques in software R & D: A literature review, in: International Conference of Software Business, Springer, 2015, pp. 139–153, https://doi.org/10.1007/978-3-319-19593-3_12.

[11] S.G. Yaman, T. Sauvola, L. Riungu-Kalliosaari, L. Hokkanen, P. Kuvaja, M. Oivo, T. Männistö, Customer involvement in continuous deployment: A systematic literature review, in: International Working Conference on Requirements Engineering: Foundation for Software Quality, Springer, 2016, pp. 249–265, https://doi.org/10.1007/978-3-319-30282-9_18.

[12] H. Holmström Olsson, H. Alahyari, J. Bosch, Climbing the "stairway to heaven"–A mulitiple-case study exploring barriers in the transition from agile development towards continuous deployment of software, in: 2012 38th Euromicro Conference on Software Engineering and Advanced Applications, IEEE, 2012, pp. 392–399, https://doi.org/10.1109/SEAA.2012.54.

[13] W.E. Deming, Out of the crisis, in: MIT Center for Advanced Engineering Study, 1986.

[14] V.R. Basili, Quantitative evaluation of software methodology, in: Proceedings of the First Pan Pacific Computer Conference, vol. 1, 1985, pp. 379–398.

[15] P. Bosch-Sijtsema, J. Bosch, User involvement throughout the innovation process in high-tech industries, J. Prod. Innov. Manage. 32 (5) (2015) 793–807, https://doi.org/10.1111/jpim.12233.

[16] G. Jurca, T.D. Hellmann, F. Maurer, Integrating agile and user-centered design: a systematic mapping and review of evaluation and validation studies of agile-ux, in: 2014 Agile Conference, IEEE, 2014, pp. 24–32, https://doi.org/10.1109/AGILE.2014.17.

[17] D. Salah, R.F. Paige, P. Cairns, A systematic literature review for agile development processes and user centred design integration, in: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, 2014, pp. 1–10, https://doi.org/10.1145/2601248.2601276.

[18] R.A. Fisher, The Design of Experiments, Oliver And Boyd, Edinburgh, London, 1937.

[19] R.K. Roy, Design of Experiments using the Taguchi Approach: 16 Steps To Product and Process Improvement, John Wiley & Sons, 2001.

[20] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell, A. Wesslén, Experimentation in Software Engineering, Springer Science & Business Media, 2012.

[21] C. Wohlin, Second-generation systematic literature studies using snowballing, in: Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering - EASE '16, ACM Press, New York, USA, 2016, pp. 1–6, https://doi.org/10.1145/2915970.2916006.

[22] X. Huang, H. Zhang, X. Zhou, M.A. Babar, S. Yang, Synthesizing qualitative research in software engineering: A critical review, in: Proceedings of the 40th International Conference on Software Engineering, ICSE '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 1207–1218, https://doi.org/10.1145/3180155.3180235.

[23] D.S. Cruzes, T. Dyba, Recommended steps for thematic synthesis in software engineering, in: 2011 International Symposium on Empirical Software Engineering and Measurement, 2011, pp. 275–284, https://doi.org/10.1109/ESEM.2011.36.

[24] R. Wieringa, N. Maiden, N. Mead, C. Rolland, Requirements engineering paper classification and evaluation criteria: a proposal and a discussion, Requir. Eng. 11 (1) (2005) 102–107, https://doi.org/10.1007/s00766-005-0021-6.

[25] D.S. Cruzes, T. Dybå, P. Runeson, M. Höst, Case studies synthesis: A thematic, cross-case, and narrative synthesis worked example, Empir. Softw. Eng. 20 (6) (2015) 1634–1665, https://doi.org/10.1007/s10664-014-9326-8.

[26] S. Easterbrook, J. Singer, M.-A. Storey, D. Damian, Selecting empirical methods for software engineering research, in: Guide to Advanced Empirical Software Engineering, Springer, London, 2008, pp. 285–311, https://doi.org/10.1007/978-1-84800-044-5_11.

[27] F. Auer, R. Ros, L. Kaltenbrunner, P. Runeson, M. Felderer, Dataset of controlled experimentation in continuous experimentation: Knowledge and challenges, 2021, https://doi.org/10.6084/m9.figshare.13712329.

[28] E. Ries, The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation To Create Radically Successful Businesses, Crown Books, 2011.

[29] C.M. Gray, Y. Kou, B. Battles, J. Hoggatt, A.L. Toombs, The dark (patterns) side of ux design, in: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, 2018, pp. 1–14, https://doi.org/10.1145/3173574.3174108.

[30] E. Engström, M.-A. Storey, P. Runeson, M. Höst, M.T. Baldassarre, How software engineering research aligns with design science: a review, Empir. Softw. Eng. 25 (4) (2020) 2630–2660.

[31] R.C. Van Nostrand, Design of experiments using the taguchi approach: 16 steps to product and process improvement, 2002.

[32] C. Flick, Informed consent and the facebook emotional manipulation study, Res. Ethics 12 (1) (2016) 14–28.

Selected Publications

[33] D.G. Feitelson, E. Frachtenberg, K.L. Beck, Development and deployment at Facebook, IEEE Internet Comput. 17 (4) (2013) 8–17, https://doi.org/10.1109/mic.2013.25.

[34] C.A. Gomez-Uribe, N. Hunt, The netflix recommender system, ACM Trans. Manage. Inf. Syst. 6 (4) (2015) 1–19, https://doi.org/10.1145/2843948.

[35] R. Kohavi, A. Deng, B. Frasca, T. Walker, Y. Xu, N. Pohlmann, Online controlled experiments at large scale, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2013, pp. 1168–1176.

[36] S. Borodovsky, S. Rosset, A/b testing at sweetim: The importance of proper statistical analysis, in: 2011 IEEE 11th International Conference on Data Mining Workshops, IEEE, 2011, pp. 733–740.

[37] R. Kohavi, T. Crook, R. Longbotham, B. Frasca, R. Henne, J.L. Ferres, T. Melamed, Online experimentation at Microsoft, Data Min. Case Stud. 11 (2009) 39.

[38] F. Fagerholm, A.S. Guinea, H. Mäenpää, J. Münch, The RIGHT model for continuous experimentation, J. Syst. Softw. 123 (2017) 292–305, https://doi.org/10.1016/j.jss.2016.03.034.

[39] S. Gupta, L. Ulanova, S. Bhardwaj, P. Dmitriev, P. Raff, A. Fabijan, The anatomy of a large-scale experimentation platform, in: 2018 IEEE International Conference on Software Architecture, ICSA, IEEE, 2018, pp. 1–109.

[40] W. Machmouchi, G. Buscher, Principles for the design of online A/B metrics, in: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR'16, ACM Press, 2016, pp. 589–590, https://doi.org/10.1145/2911451.2926731.

[41] U. Eklund, J. Bosch, Architecture for large-scale innovation experiment systems, in: 2012 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture, IEEE, 2012, pp. 244–248.

[42] R. Kohavi, R. Longbotham, D. Sommerfield, R.M. Henne, Controlled experiments on the web: Survey and practical guide, Data Min. Knowl. Discov. 18 (1) (2008) 140–181, https://doi.org/10.1007/s10618-008-0114-1.

[43] B. Fitzgerald, K.-J. Stol, Continuous software engineering: A roadmap and agenda, J. Syst. Softw. 123 (2017) 176–189, https://doi.org/10.1016/j.jss.2015.06.063.

[44] E. Lindgren, J. Münch, Raising the odds of success: The current state of experimentation in product development, Inf. Softw. Technol. 77 (2016) 80–91, https://doi.org/10.1016/j.infsof.2016.04.008.

[45] B. Letham, B. Karrer, G. Ottoni, E. Bakshy, Constrained bayesian optimization with noisy experiments, Bayesian Anal. 14 (2) (2019) 495–519, https://doi.org/10.1214/18-ba1110.

[46] R. Ros, E. Bjarnason, Continuous experimentation scenarios: A case study in e-commerce, in: 2018 44th Euromicro Conference on Software Engineering and Advanced Applications, SEAA, IEEE, 2018, pp. 353–356.

[47] H.H. Olsson, J. Bosch, From opinions to data-driven software R & D: A multi-case study on how to close the 'open loop' problem, in: 2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications, IEEE, 2014, pp. 9–16, https://doi.org/10.1109/seaa.2014.75.

[48] K. Kevic, B. Murphy, L. Williams, J. Beckmann, Characterizing experimentation in continuous deployment: a case study on bing, in: 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track, ICSE-SEIP, IEEE, 2017, pp. 123–132.

[49] T. Sauvola, M. Kelanti, J. Hyysalo, P. Kuvaja, K. Liukkunen, Continuous improvement and validation with customer touchpoint model in software development, in: Proceedings of the 13th International Conference on Software Engineering Advances - ICSEA '18, IARIA XPS Press, 2018, p. 62.

[50] X. Amatriain, Beyond data: From user information to business value through personalized recommendations and consumer science, in: Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management - CIKM '13, 2013, pp. 2201–2208, https://doi.org/10.1145/2505515.2514701.

[51] A. Fabijan, P. Dmitriev, H.H. Olsson, J. Bosch, The evolution of continuous experimentation in software product development: from data to a data-driven organization at scale, in: 2017 IEEE/ACM 39th International Conference on Software Engineering, ICSE, IEEE, 2017, pp. 770–780.

[52] A. Fabijan, P. Dmitriev, H.H. Olsson, J. Bosch, Online controlled experimentation at scale: an empirical survey on the current state of a/b testing, in: 2018 44th Euromicro Conference on Software Engineering and Advanced Applications, SEAA, IEEE, 2018, pp. 68–72.

[53] A. Fabijan, P. Dmitriev, H.H. Olsson, J. Bosch, Effective online controlled experiment analysis at large scale, in: 2018 44th Euromicro Conference on Software Engineering and Advanced Applications, SEAA, IEEE, 2018, pp. 64–67.

[54] D.I. Mattos, P. Dmitriev, A. Fabijan, J. Bosch, H.H. Olsson, An activity and metric model for online controlled experiments, in: Product-Focused Software Process Improvement, Springer International Publishing, 2018, pp. 182–198, https://doi.org/10.1007/978-3-030-03673-7_14.

[55] G. Schermann, J. Cito, P. Leitner, U. Zdun, H.C. Gall, We're doing it live: A multi-method empirical study on continuous experimentation, Inf. Softw. Technol. 99 (2018) 41–57, https://doi.org/10.1016/j.infsof.2018.02.010.

[56] D.I. Mattos, J. Bosch, H.H. Olsson, Your system gets better every day you use it: Towards automated continuous experimentation, in: 2017 43rd Euromicro Conference on Software Engineering and Advanced Applications, SEAA, IEEE, 2017, pp. 256–265, https://doi.org/10.1109/seaa.2017.15.

[57] G. Schermann, D. Schöni, P. Leitner, H.C. Gall, Bifrost – supporting continuous deployment with automated enactment of multi-phase live testing strategies, in: Proceedings of the 17th International Middleware Conference on - Middleware '16, ACM Press, 2016, pp. 1–14, https://doi.org/10.1145/2988336.2988348.

[58] K. Koukouvis, R.A. Cubero, P. Pelliccione, A/B testing in e-commerce sales processes, in: Lecture Notes in Computer Science, Springer International Publishing, 2016, pp. 133–148, https://doi.org/10.1007/978-3-319-45892-2_10.

[59] S. Firmenich, A. Garrido, J. Grigera, J.M. Rivero, G. Rossi, Usability improvement through A/B testing and refactoring, Softw. Qual. J. 27 (1) (2018) 203–240, https://doi.org/10.1007/s11219-018-9413-y.

[60] Y. Xu, N. Chen, A. Fernandez, O. Sinno, A. Bhasin, From infrastructure to culture: A/B testing challenges in large scale social networks, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15, ACM Press, 2015, pp. 2227–2236, https://doi.org/10.1145/2783258.2788602.

[61] J. Björklund, J. Ljungblad, J. Bosch, Lean product development in early stage startups, in: CEUR Worshop Proceedings: From Start-ups to SaaS Conglomerate - Life Cycles of Software Products Workshop - IW-LCSP '13, 2013, pp. 19–32.

[62] M. Gutbrod, J. Münch, M. Tichy, How do software startups approach experimentation? Empirical results from a qualitative interview study, in: Product-Focused Software Process Improvement, Springer International Publishing, 2017, pp. 297–304, https://doi.org/10.1007/978-3-319-69926-4_21.

[63] O. Rissanen, J. Münch, Continuous experimentation in the b2b domain: a case study, in: 2015 IEEE/ACM 2nd International Workshop on Rapid Continuous Software Engineering, IEEE, 2015, pp. 12–18.

[64] S. Yaman, T. Mikkonen, R. Suomela, Continuous experimentation in mobile game development, in: 2018 44th Euromicro Conference on Software Engineering and Advanced Applications, SEAA, IEEE, 2018, pp. 345–352.

[65] H. Xie, J. Aurisset, Improving the sensitivity of online controlled experiments, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16, ACM Press, 2016, pp. 645–654, https://doi.org/10.1145/2939672.2939733.

[66] A. Deng, Objective bayesian two sample hypothesis testing for online controlled experiments, in: Proceedings of the 24th International Conference on World Wide Web - WWW '15 Companion, ACM Press, 2015, pp. 923–928, https://doi.org/10.1145/2740908.2742563.

[67] A. Deng, Y. Xu, R. Kohavi, T. Walker, Improving the sensitivity of online controlled experiments by utilizing pre-experiment data, in: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining - WSDM '13, ACM Press, 2013, pp. 123–132, https://doi.org/10.1145/2433396.2433413.

[68] H. Hohnhold, D. O'Brien, D. Tang, Focusing on the long-term: It's good for users and business, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15, ACM Press, 2015, pp. 1849–1858, https://doi.org/10.1145/2783258.2788583.

[69] N. Appiktala, M. Chen, M. Natkovich, J. Walters, Demystifying dark matter for online experimentation, in: 2017 IEEE International Conference on Big Data, Big Data, IEEE, 2017, pp. 1620–1626, https://doi.org/10.1109/bigdata.2017.8258096.

[70] N. Chen, M. Liu, Y. Xu, How A/B tests could go wrong: Automatic diagnosis of invalid online experiments, in: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining - WSDM '19, ACM Press, 2019, pp. 501–509, https://doi.org/10.1145/3289600.3291000.

[71] M. Nolting, J.E. von Seggern, Context-based A/B test validation, in: Proceedings of the 25th International Conference Companion on World Wide Web - WWW '16 Companion, ACM Press, 2016, pp. 277–278, https://doi.org/10.1145/2872518.2889306.

[72] W. Duivesteijn, T. Farzami, T. Putman, E. Peer, H.J.P. Weerts, J.N. Adegeest, G. Foks, M. Pechenizkiy, Have it both ways—from A/B testing to A & B testing with exceptional model mining, in: Machine Learning and Knowledge Discovery in Databases, Springer International Publishing, 2017, pp. 114–126, https://doi.org/10.1007/978-3-319-71273-4_10.

[73] Y. Xie, N. Chen, X. Shi, False discovery rate controlled heterogeneous treatment effect detection for online controlled experiments, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD '18, ACM Press, 2018, pp. 876–885, https://doi.org/10.1145/3219819.3219860.

[74] A. Deng, X. Shi, Data-driven metric development for online controlled experiments: Seven lessons learned, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16, ACM Press, 2016, pp. 77–86, https://doi.org/10.1145/2939672.2939700.

[75] P. Dmitriev, X. Wu, Measuring metrics, in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management - CIKM '16, ACM Press, 2016, pp. 429–437, https://doi.org/10.1145/2983323.2983356.

[76] A. Drutsa, A. Ufland, G. Gusev, Practical aspects of sensitivity in online experimentation with user engagement metrics, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management - CIKM '15, ACM Press, 2015, pp. 763–772, https://doi.org/10.1145/2806416.2806496.

[77] R. Budylin, A. Drutsa, I. Katsev, V. Tsoy, Consistent transformation of ratio metrics for efficient online controlled experiments, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining - WSDM '18, ACM Press, 2018, pp. 55–63, https://doi.org/10.1145/3159652.3159699.

[78] A. Drutsa, G. Gusev, P. Serdyukov, Using the delay in a treatment effect to improve sensitivity and preserve directionality of engagement metrics in a/b experiments, in: Proceedings of the 26th International Conference on World Wide Web - WWW '17, ACM Press, 2017, pp. 1301–1310, https://doi.org/10.1145/3038912.3052664.

[79] E. Kharitonov, A. Drutsa, P. Serdyukov, Learning sensitive combinations of a/b test metrics, in: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining - WSDM '17, ACM Press, 2017, pp. 651–659, https://doi.org/10.1145/3018661.3018708.

[80] W. Machmouchi, A.H. Awadallah, I. Zitouni, G. Buscher, Beyond success rate: Utility as a search quality metric for online experiments, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management - CIKM '17, ACM Press, 2017, pp. 757–765, https://doi.org/10.1145/3132847.3132850.

[81] A. Drutsa, Sign-aware periodicity metrics of user engagement for online search quality evaluation, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '15, ACM, Press, 2015, pp. 845–854, https://doi.org/10.1145/2766462.2767814.

[82] A. Drutsa, G. Gusev, P. Serdyukov, Future user engagement prediction and its application to improve the sensitivity of online experiments, in: Proceedings of the 24th International Conference on World Wide Web - WWW '15, ACM Press, 2015, pp. 256–266, https://doi.org/10.1145/2736277.2741116.

[83] A. Poyarkov, A. Drutsa, A. Khalyavin, G. Gusev, P. Serdyukov, Boosted decision tree regression adjustment for variance reduction in online controlled experiments, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16, ACM Press, 2016, pp. 235–244, https://doi.org/10.1145/2939672.2939688.

[84] A. Deng, T. Li, Y. Guo, Statistical inference in two-stage online controlled experiments with treatment selection and validation, in: Proceedings of the 23rd International Conference on World Wide Web - WWW '14, ACM Press, 2014, pp. 609–618, https://doi.org/10.1145/2566486.2568028.

[85] A. Peysakhovich, D. Eckles, Learning causal effects from many randomized experiments using regularized instrumental variables, in: Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18, ACM Press, 2018, pp. 699–707, https://doi.org/10.1145/3178876.3186151.

[86] R. Kohavi, A. Deng, R. Longbotham, Y. Xu, Seven rules of thumb for web site experimenters, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '14, ACM Press, 2014, pp. 1857–1866, https://doi.org/10.1145/2623330.2623341.

[87] E. Claeys, P. Gançarski, M. Maumy-Bertrand, H. Wassner, Regression tree for bandits models in A/B testing, in: Advances in Intelligent Data Analysis XVI, Springer International Publishing, 2017, pp. 52–62, https://doi.org/10.1007/978-3-319-68765-0_5.

[88] D.N. Hill, H. Nassif, Y. Liu, A. Iyer, S. Vishwanathan, An efficient bandit algorithm for realtime multivariate optimization, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '17, ACM Press, 2017, pp. 1813–1821, https://doi.org/10.1145/3097983.3098184.

[89] D.I. Mattos, E. Mårtensson, J. Bosch, H.H. Olsson, Optimization experiments in the continuous space, in: Search-Based Software Engineering, Springer International Publishing, 2018, pp. 293–308, https://doi.org/10.1007/978-3-319-99241-9_16.

[90] R. Miikkulainen, G. Lamba, N. Iscoe, A. Shagrin, R. Cordell, S. Nazari, C. Schoolland, M. Brundage, J. Epstein, R. Dean, Conversion rate optimization through evolutionary computation, in: Proceedings of the Genetic and Evolutionary Computation Conference on - GECCO '17, ACM Press, 2017, pp. 1193–1199, https://doi.org/10.1145/3071178.3071312.

[91] R. Ros, E. Bjarnason, P. Runeson, Automated controlled experimentation on software by evolutionary bandit optimization, in: Search Based Software Engineering, Springer International Publishing, 2017, pp. 190–196, https://doi.org/10.1007/978-3-319-66299-2_18.

[92] G. Tamburrelli, A. Margara, Towards automated A/B testing, in: Search-Based Software Engineering, Springer International Publishing, 2014, pp. 184–198, https://doi.org/10.1007/978-3-319-09940-8_13.

[93] J. Bosch, Speed, data, and ecosystems: The future of software engineering, IEEE Softw. 33 (1) (2016) 82–88, https://doi.org/10.1109/ms.2016.14.

[94] M. Speicher, A. Both, M. Gaedke, Ensuring web interface quality through usability-based split testing, in: Lecture Notes in Computer Science, Springer International Publishing, 2014, pp. 93–110, https://doi.org/10.1007/978-3-319-08245-5_6.

[95] D.N. Hill, R. Moakler, A.E. Hubbard, V. Tsemekhman, F. Provost, K. Tsemekhman, Measuring causal impact of online actions via natural experiments: Application to display advertising, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15, ACM Press, 2015, pp. 1839–1847, https://doi.org/10.1145/2783258.2788622.

[96] Y. Xu, N. Chen, Evaluating mobile apps with A/B and quasi A/B tests, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16, ACM Press, 2016, pp. 313–322, https://doi.org/10.1145/2939672.2939703.

[97] S. Iitsuka, Y. Matsuo, Website optimization problem and its solutions, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15, ACM Press, 2015, pp. 447–456, https://doi.org/10.1145/2783258.2783351.

[98] I. Gerostathopoulos, A.N. Uysal, C. Prehofer, T. Bures, A tool for online experiment-driven adaptation, in: 2018 IEEE 3rd International Workshops on Foundations and Applications of Self∗ Systems (FAS∗W), IEEE, 2018, pp. 100–105, https://doi.org/10.1109/fas-w.2018.00032.

[99] E. Kharitonov, C. Macdonald, P. Serdyukov, I. Ounis, Optimised scheduling of online experiments, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '15, ACM Press, 2015, pp. 453–462, https://doi.org/10.1145/2766462.2767706.

[100] G. Schermann, P. Leitner, Search-based scheduling of experiments in continuous deployment, in: 2018 IEEE International Conference on Software Maintenance and Evolution (ICSME), IEEE, 2018, pp. 485–495.

[101] J. Cámara, A. Kobsa, Facilitating controlled tests of website design changes: A systematic approach, in: Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2009, pp. 370–378, https://doi.org/10.1007/978-3-642-02818-2_30.

[102] E. Bakshy, D. Eckles, M.S. Bernstein, Designing and deploying online field experiments, in: Proceedings of the 23rd International Conference on World Wide Web - WWW '14, ACM Press, 2014, pp. 283–292, https://doi.org/10.1145/2566486.2567967.

[103] D. Tang, A. Agarwal, D. O'Brien, M. Meyer, Overlapping experiment infrastructure: More, better, faster experimentation, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '10, ACM Press, 2010, pp. 17–26, https://doi.org/10.1145/1835804.1835810.

[104] M.T. Rahman, L.-P. Querel, P.C. Rigby, B. Adams, Feature toggles, in: Proceedings of the 13th International Workshop on Mining Software Repositories - MSR '16, ACM Press, 2016, pp. 201–211, https://doi.org/10.1145/2901739.2901745.

[105] D. Westermann, J. Happe, R. Farahbod, An experiment specification language for goal-driven, automated performance evaluations, in: Proceedings of the 28th Annual ACM Symposium on Applied Computing - SAC '13, ACM Press, 2013, pp. 1043–1048, https://doi.org/10.1145/2480362.2480561.

[106] E. Bakshy, D. Eckles, Uncertainty in online experiments with dependent data: An evaluation of bootstrap methods, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '13, ACM Press, 2013, pp. 1303–1311, https://doi.org/10.1145/2487575.2488218.

[107] Z. Zhao, M. Chen, D. Matheson, M. Stone, Online experimentation diagnosis and troubleshooting beyond AA validation, in: 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), IEEE, 2016, pp. 498–507, https://doi.org/10.1109/dsaa.2016.61.

[108] R. Chen, M. Chen, M.R. Jadav, J. Bae, D. Matheson, Faster online experimentation by eliminating traditional A/A validation, in: 2017 IEEE International Conference on Big Data (Big Data), IEEE, 2017, pp. 1635–1641, https://doi.org/10.1109/bigdata.2017.8258098.

[109] A. Deng, V. Hu, Diluted treatment effect estimation for trigger analysis in online controlled experiments, in: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining - WSDM '15, ACM Press, 2015, pp. 349–358, https://doi.org/10.1145/2684822.2685307.

[110] A. Das, H. Ranganath, When web personalization misleads bucket testing, in: Proceedings of the 1st Workshop on User Engagement Optimization - UEO '13, ACM Press, 2013, pp. 17–20, https://doi.org/10.1145/2512875.2512879.

[111] J. Gupchup, Y. Hosseinkashi, P. Dmitriev, D. Schneider, R. Cutler, A. Jefremov, M. Ellis, Trustworthy experimentation under telemetry loss, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management - CIKM '18, ACM Press, 2018, pp. 387–396, https://doi.org/10.1145/3269206.3271747.

[112] M.R. Lee, M. Shen, Winner's curse: Bias estimation for total effects of features in online controlled experiments, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD '18, ACM Press, 2018, pp. 491–499, https://doi.org/10.1145/3219819.3219905.

[113] A.M. Medina, S. Vassilvitskii, D. Yin, Online learning for non-stationary A/B tests, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management - CIKM '18, ACM Press, 2018, pp. 317–326, https://doi.org/10.1145/3269206.3271718.

[114] Y. Xu, W. Duan, S. Huang, SQR: Balancing speed, quality and risk in online experiments, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD '18, ACM Press, 2018, pp. 895–904, https://doi.org/10.1145/3219819.3219875.

[115] A. Deng, J. Lu, S. Chen, Continuous monitoring of A/B tests without pain: Optional stopping in bayesian testing, in: 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), IEEE, 2016, pp. 243–252, https://doi.org/10.1109/dsaa.2016.33.

[116] E. Kharitonov, A. Vorobev, C. Macdonald, P. Serdyukov, I. Ounis, Sequential testing for early stopping of online experiments, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '15, ACM Press, 2015, pp. 473–482, https://doi.org/10.1145/2766462.2767729.

[117] R. Johari, P. Koomen, L. Pekelis, D. Walsh, Peeking at A/B tests: Why it matters, and what to do about it, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '17, ACM Press, 2017, pp. 1517–1525, https://doi.org/10.1145/3097983.3097992.

[118] V. Abhishek, S. Mannor, A nonparametric sequential test for online randomized experiments, in: Proceedings of the 26th International Conference on World Wide Web Companion - WWW '17 Companion, ACM Press, 2017, pp. 610–616, https://doi.org/10.1145/3041021.3054196.

[119] N. Ju, D. Hu, A. Henderson, L. Hong, A sequential test for selecting the better variant: Online A/B testing, adaptive allocation, and continuous monitoring, in: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining - WSDM '19, ACM Press, 2019, pp. 492–500, https://doi.org/10.1145/3289600.3291025.

[120] J. Bosch, Building products as innovation experiment systems, in: Lecture Notes in Business Information Processing, Springer Berlin Heidelberg, 2012, pp. 27–39, https://doi.org/10.1007/978-3-642-30746-1_3.

[121] S.G. Yaman, M. Munezero, J. Münch, F. Fagerholm, O. Syd, M. Aaltola, C. Palmu, T. Männistö, Introducing continuous experimentation in large software-intensive product and service organisations, J. Syst. Softw. 133 (2017) 195–211, https://doi.org/10.1016/j.jss.2017.07.009.

[122] H.H. Olsson, J. Bosch, A. Fabijan, Experimentation that matters: A multi-case study on the challenges with A/B testing, in: Lecture Notes in Business Information Processing, Springer International Publishing, 2017, pp. 179–185, https://doi.org/10.1007/978-3-319-69191-6_12.

[123] P. Dmitriev, S. Gupta, D.W. Kim, G. Vaz, A dirty dozen: Twelve common metric interpretation pitfalls in online controlled experiments, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '17, ACM Press, 2017, pp. 1427–1436, https://doi.org/10.1145/3097983.3098024.

[124] M. Conti, A. Gangwal, S.P. Gochhayat, G. Tolomei, Spot the difference: Your bucket is leaking : A novel methodology to expose A/B testing effortlessly, in: 2018 IEEE Conference on Communications and Network Security (CNS), IEEE, 2018, pp. 1–7, https://doi.org/10.1109/cns.2018.8433122.

[125] T. Crook, B. Frasca, R. Kohavi, R. Longbotham, 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 - KDD '09, ACM Press, 2009, pp. 1105–1114, https://doi.org/10.1145/1557019.1557139.

[126] P. Dmitriev, B. Frasca, S. Gupta, R. Kohavi, G. Vaz, Pitfalls of long-term online controlled experiments, in: 2016 IEEE International Conference on Big Data (Big Data), IEEE, 2016, pp. 1367–1376, https://doi.org/10.1109/bigdata.2016.7840744.

[127] R. Kohavi, A. Deng, B. Frasca, R. Longbotham, T. Walker, Y. Xu, Trustworthy online controlled experiments: Five puzzling outcomes explained, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '12, ACM Press, 2012, pp. 786–794, https://doi.org/10.1145/2339530.2339653.

[128] L. Lu, C. Liu, Separation strategies for three pitfalls in A/B testing, in: The Second Workshop on User Engagement Optimization (UEO) At KDD '14, ACM, 2014, p. 1.

[129] S. Yaman, F. Fagerholm, M. Munezero, H. Maenpaa, T. Mannisto, Notifying and involving users in experimentation: Ethical perceptions of software practitioners, in: 2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), IEEE, 2017, pp. 199–204, https://doi.org/10.1109/esem.2017.31.

[130] S. Jiang, J. Martin, C. Wilson, Who's the guinea pig?: Investigating online A/B/N tests in-the-wild, in: Proceedings of the Conference on Fairness, Accountability, and Transparency - FAT∗ '19, ACM Press, 2019, pp. 201–210, https://doi.org/10.1145/3287560.3287565.

[131] F. Lettner, C. Holzmann, P. Hutflesz, Enabling A/B testing of native mobile applications by remote user interface exchange, in: Computer Aided Systems Theory - EUROCAST 2013, Springer Berlin Heidelberg, 2013, pp. 458–466, https://doi.org/10.1007/978-3-642-53862-9_58.

[132] J. Bosch, H.H. Olsson, Data-driven continuous evolution of smart systems, in: Proceedings of the 11th International Workshop on Software Engineering for Adaptive and Self-Managing Systems - SEAMS '16, ACM Press, 2016, pp. 28–34, https://doi.org/10.1145/2897053.2897066.

[133] F. Giaimo, C. Berger, C. Kirchner, Considerations about continuous experimentation for resource-constrained platforms in self-driving vehicles, in: Software Architecture, Springer International Publishing, 2017, pp. 84–91, https://doi.org/10.1007/978-3-319-65831-5_6.

[134] D.I. Mattos, J. Bosch, H.H. Olsson, Challenges and strategies for undertaking continuous experimentation to embedded systems: Industry and research perspectives, in: Lecture Notes in Business Information Processing, Springer International Publishing, 2018, pp. 277–292, https://doi.org/10.1007/978-3-319-91602-6_20.

[135] F.G. Azevedo, B.D. Nogueira, F. Murai, A.P.C. Silva, Estimation errors in network A/B testing due to sample variance and model misspecification, in: 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI), IEEE, 2018, pp. 540–545, https://doi.org/10.1109/wi.2018.00-40.

[136] L. Backstrom, J. Kleinberg, Network bucket testing, in: Proceedings of the 20th International Conference on World Wide Web - WWW '11, ACM Press, 2011, pp. 615–624, https://doi.org/10.1145/1963405.1963492.

[137] D. Choi, Estimation of monotone treatment effects in network experiments, J. Amer. Statist. Assoc. 112 (519) (2017) 1147–1155, https://doi.org/10.1080/01621459.2016.1194845.

[138] A. Goswami, W. Han, Z. Wang, A. Jiang, Controlled experiments for decision-making in e-Commerce search, in: 2015 IEEE International Conference on Big Data (Big Data), IEEE, 2015, pp. 1094–1102, https://doi.org/10.1109/bigdata.2015.7363863.

[139] J. Wang, D. Goldberg, P. Burke, D. Bhoite, Designing and analyzing A/B tests in an online marketplace, in: 2018 IEEE International Conference on Data Mining Workshops (ICDMW), IEEE, 2018, pp. 1447–1452, https://doi.org/10.1109/icdmw.2018.00206.

[140] D. Coey, M. Bailey, People and cookies: Imperfect treatment assignment in online experiments, in: Proceedings of the 25th International Conference on World Wide Web - WWW '16, ACM Press, 2016, pp. 1103–1111, https://doi.org/10.1145/2872427.2882984.

[141] A. Deng, J. Lu, J. Litz, Trustworthy analysis of online a/b tests: Pitfalls, challenges and solutions, in: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining - WSDM '17, ACM Press, 2017, pp. 641–649, https://doi.org/10.1145/3018661.3018677.

[142] R. Kohavi, R. Longbotham, Unexpected results in online controlled experiments, ACM SIGKDD Explorations Newslett. 12 (2) (2011) 31, https://doi.org/10.1145/1964897.1964905.

[143] M. Adinata, I. Liem, A/B test tools of native mobile application, in: 2014 International Conference on Data and Software Engineering (ICODSE), IEEE, 2014, pp. 1–6, https://doi.org/10.1109/icodse.2014.7062683.

[144] J. Bosch, U. Eklund, Eternal embedded software: Towards innovation experiment systems, in: Leveraging Applications of Formal Methods, Verification and Validation. Technologies for Mastering Change, Springer Berlin Heidelberg, 2012, pp. 19–31, https://doi.org/10.1007/978-3-642-34026-0_3.

[145] F. Giaimo, H. Yin, C. Berger, I. Crnkovic, Continuous experimentation on cyber-physical systems – challenges and opportunities, in: Proceedings of the Scientific Workshop Proceedings of XP2016 on - XP '16 Workshops, ACM Press, 2016, pp. 1–2, https://doi.org/10.1145/2962695.2962709.

[146] I. Gerostathopoulos, D. Skoda, F. Plasil, T. Bures, A. Knauss, Architectural homeostasis in self-adaptive software-intensive cyber-physical systems, in: Software Architecture, Springer International Publishing, 2016, pp. 113–128, https://doi.org/10.1007/978-3-319-48992-6_8.

[147] I. Gerostathopoulos, C. Prehofer, T. Bures, Adapting a system with noisy outputs with statistical guarantees, in: Proceedings of the 13th International Conference on Software Engineering for Adaptive and Self-Managing Systems - SEAMS '18, ACM Press, 2018, pp. 998–1001, https://doi.org/10.1145/3194133.3194152.

[148] I. Gerostathopoulos, C. Prehofer, L. Bulej, T. Bureš, V. Horký, P. Tuma, Cost-aware stage-based experimentation: challenges and emerging results, in: 2018 IEEE International Conference on Software Architecture Companion (ICSA-C), IEEE, 2018, pp. 72–75.

[149] T. Buchert, C. Ruiz, L. Nussbaum, O. Richard, A survey of general-purpose experiment management tools for distributed systems, Future Gener. Comput. Syst. 45 (2015) 1–12, https://doi.org/10.1016/j.future.2014.10.007.

[150] D. Jayasinghe, J. Kimball, S. Choudhary, T. Zhu, C. Pu, An automated approach to create, store, and analyze large-scale experimental data in clouds, in: 2013 IEEE 14th International Conference on Information Reuse & Integration (IRI), IEEE, 2013, pp. 357–364, https://doi.org/10.1109/iri.2013.6642493.

[151] L. Katzir, E. Liberty, O. Somekh, Framework and algorithms for network bucket testing, in: Proceedings of the 21st International Conference on World Wide Web - WWW '12, ACM Press, 2012, pp. 1029–1036, https://doi.org/10.1145/2187836.2187974.

[152] H. Gui, Y. Xu, A. Bhasin, J. Han, Network A/B testing, in: Proceedings of the 24th International Conference on World Wide Web - WWW '15, ACM Press, 2015, pp. 399–409, https://doi.org/10.1145/2736277.2741081.

[153] M. Saveski, J. Pouget-Abadie, G. Saint-Jacques, W. Duan, S. Ghosh, Y. Xu, E.M. Airoldi, Detecting network effects, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '17, ACM Press, 2017, pp. 1027–1035, https://doi.org/10.1145/3097983.3098192.

[154] D. Eckles, B. Karrer, J. Ugander, Design and analysis of experiments in networks: Reducing bias from interference, J. Causal Inference 5 (1) (2016) https://doi.org/10.1515/jci-2015-0021.

[155] A. Fabijan, P. Dmitriev, H.H. Olsson, J. Bosch, The benefits of controlled experimentation at scale, in: 2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA), IEEE, 2017, pp. 18–26.

[156] A. Fabijan, P. Dmitriev, H.H. Olsson, J. Bosch, The online controlled experiment lifecycle, IEEE Softw. (2018) 1, https://doi.org/10.1109/ms.2018.2875842.

[157] D. Liu, S. Xu, B. Zhang, C. Wang, C. Li, F. Zhou, Issues with conducting controlled on-line experiments for e-commerce, in: 2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), IEEE, 2017, pp. 187–192, https://doi.org/10.1109/snpd.2017.8022721.

[158] D.I. Mattos, J. Bosch, H.H. Olsson, More for less: Automated experimentation in software-intensive systems, in: Product-Focused Software Process Improvement, Springer International Publishing, 2017, pp. 146–161, https://doi.org/10.1007/978-3-319-69926-4_12.

[159] S. Chasins, P.M. Phothilimthana, Dicer, in: Proceedings of the 24th International Conference on World Wide Web - WWW '15 Companion, ACM Press, 2015, pp. 1321–1326, https://doi.org/10.1145/2740908.2741699.

[160] H.H. Olsson, J. Bosch, H. Alahyari, Towards R & D as innovation experiment systems: A framework for moving beyond agile software development, in: Proceedings of the IASTED, ACTAPRESS, 2013, pp. 798–805, https://doi.org/10.2316/P.2013.796-008.