A Literature Study to Characterize Continuous Experimentation in Software Engineering

Vladimir M. Erthal[1], Bruno P. de Souza[1], Paulo Sérgio M. dos Santos[2], Guilherme H. Travassos[1]

[1]PESC/COPPE – Federal University of Rio de Janeiro (UFRJ)
Rio de Janeiro – RJ – Brazil

[2]Department of Applied Informatics – Federal University of the State of Rio de Janeiro
Rio de Janeiro – RJ – Brazil

[1]{vladimirerthal, bpsouza, ght}@cos.ufrj.br, [2]pasemes@uniriotec.br

Abstract. *Continuous Experimentation (CE) has become increasingly popular across industry and academic communities. Given its rapid evolution in software engineering (SE), the lack of a common understanding of CE can jeopardize new implementations and justify research efforts. Therefore, this literature study characterizes CE in SE based on its definitions, processes, and strategies for experimentation available in the technical literature. Seventy-six sources of information provided many different definitions, processes, and experimental procedures used to describe CE in SE. Despite the increasing use of CE in SE, it is impossible to observe a common terminology yet to support its characterization and use.*

1. Introduction

Continuous Experimentation (CE) has emerged as a new development practice for software systems. It aims to support software systems engineering through a systematic definition of *hypotheses*, continuous delivery to the end-users, and monitoring metrics to assess the acceptance or rejection of *hypotheses* based on evidence of actual use [Fagerholm et al. 2017]. It arose from the context of agile practices with a strong basis on the *Lean Startup* methodology of "build-measure-learn" [Fagerholm et al. 2014].

CE plays a fundamental role in supporting collaboration between the customer and the development team to discover new software requirements [Olsson and Bosch 2013a]. Also, using controlled experiments, CE guides software development organizations to evaluate and prioritize their development efforts, such as implementing a specific requirement/feature based on users' data [Olsson and Bosch 2014] and discovering the real needs of the users to create value and innovation in the product. It has been widely implemented by several big companies, mainly in web and mobile software systems. Large organizations such as Facebook, Google, Microsoft, LinkedIn, and Netflix have reported their experience using CE to evolve their software products [Auer et al. 2021].

Beyond CE, different expressions have been proposed to refer to this practice in the technical literature, such as Data-Driven Development (DDD) [Bosch and Olsson 2017], Innovation Experiment System (IES) [Bosch 2012], Online Controlled Experiments (OCE) [Kohavi et al. 2013], among others. As a result, the studies report

countless processes and strategies from the vast array of expressions [Auer et al., 2021]. Therefore, it is not easy to observe a common understanding regarding continuous experimentation in software development. Moreover, these multiple understandings and perspectives make it difficult to synthesize the papers' contributions. Sometimes, it uses different expressions of the same concept or other words for different concepts. The lack of a common terminology motivated us to understand better and consolidate CE's definitions, processes, and experiment strategies. We believe it can contribute to the technical discussion regarding CE in software engineering (SE) by providing a common knowledge base. Therefore, we performed a literature study to identify and characterize its different definitions, processes, and experimental strategies to support the initial discussions towards organizing a common terminology of continuous experimentation in software engineering.

Besides this introduction, this paper offers the following parts. Section 2 describes the related work used as a seed for this investigation. Next, section 3 introduces the literature study protocol. Section 4 reports the main findings regarding CE's definitions, strategies, and processes. Next, section 5 discusses such findings and presents the implications of CE in SE. Section 6 shows the threats to validity. Finally, section 7 concludes by suggesting some future actions.

2. Related Works

Continuous experimentation has been an object of secondary studies since 2018, as it can be observed in six secondary studies dedicated to this topic. First, Auer and Felderer (2018) published an extensive systematic mapping of 82 primary studies. They addressed questions such as the amount of research activity, the intensity of collaboration between industry and academia, the kind of contributions provided, the most frequently investigated research topics, and the terms used for CE. In the same year, Ros and Runeson (2018) also published a mapping study with 62 primary sources regarding questions focused on the main topics researched within CE, the kind of organizations that use CE, and the characteristics of the experiments that have been used with CE. Also, in 2018, Mattos, Bosch, and Olsson (2018) published a literature review on 42 papers, but in this case, with a focus on CE adoption by embedded systems.

In 2020, Auer, Lee, and Felderer (2020) performed a secondary study with 14 papers focused on experiment characteristics. They proposed a taxonomy for creating experiments used in CE, seeking characteristics, guidelines, checklists, and review processes. Giaimo, Andrade, and Berger (2020) published a literature review with eight papers focused on applying CE in cyber-physical systems in the same year. Finally, in 2021, Auer, Ros, Kaltenbrunner, Runeson, and Felderer (2021) published a systematic literature review on 128 papers addressing three questions: the core constituents of a CE framework, the experiment strategies (that they call technical solutions) applied within CE and its challenges and benefits.

These secondary studies [Auer and Felderer 2018] [Ros and Runeson, 18] [Olsson, 18] [Auer et al., 20] [Giaimo et al., 20] [Auer et al., 21] define CE in different ways. It evidences the lack of a common definition of CE. These studies use CE as a general term, encompassing different perspectives, such as Data-Driven Development, Online Controlled Experiments, and Innovation Experiment Systems. As far as we could experience on our software projects, even though these different terms can share

common practices, they do not represent the same concept. Therefore, they can blur the perspectives of practitioners and researchers when selecting CE practices that best apply to their specific software system projects.

3. Literature Study

3.1. Planning

Previous secondary studies (see Section 2) investigated CE, although with different goals from our study. They identified many primary sources of information. Therefore, we understood that a search strategy based on database searches was unnecessary. Thus, we first executed an *ad-hoc* search, including the secondary studies, to create the seed to perform our literature study using the Snowballing technique. Further, for the remaining phases of the review, we followed the practices of literature studies in software engineering as suggested in [Kuhrmann et al. 2017] for replicability and auditing of the results. It includes defining appropriate research questions, a search string, inclusion and exclusion criteria, data collection, dataset cleaning, and study selection.

Following the GQM (Goal/Question/Metric) paradigm [Basili et al. 1794], this study aims *to analyze* Continuous Experimentation practice *with the purpose of* characterizing its definitions, common expressions, processes, models, and experiment strategies *from the point of view of* SE researchers *in the context of* the SE technical literature provided by the Scopus database and snowballing. The Research Questions (RQs) detail the main aspects of the investigation [Table 1].

The term "models" is any graphical representation of any part of the experimentation approach, such as processes, frameworks, lifecycles, and architectures. This definition was used to compare the papers' organizational practices suggested or reported. Similarly, "experiment strategies" refer to obtaining and analyzing the user data needed to conduct the experiments in a CE process.

The initial *ad-hoc* search was performed using the Scopus database, a stable and large coverage search engine, basing the search on widespread expressions and limiting the search period for the last six years (2015 to 2021). The inclusion and exclusion criteria allowed us to get acquainted with the literature and create the seed for snowballing. Table 1 shows all these features. The snowballing technique searches for a research theme related to the initial articles by looking at those referenced by the initial set (backward) and those referred to (forward) [Wöhlin 2014]. The same inclusion and exclusion criteria supported the decision on the suitability of the sources of information.

Table 1. Summary of the literature study.

Research Questions	RQ1: Which are the expressions and definitions associated with the CE concept?
	RQ2: Which are the processes used for CE?
	RQ3: Which are the experiment strategies used for conducting CE?
Search String	("continuous experimentation" OR "continuous software experimentation" OR "experiment systems" OR "data-driven development" OR "A/B tests" OR "A/B testing" OR "online controlled experiments" OR "online controlled experimentation" OR "innovation experiment system" OR "Experiment-driven software development")
Inclusion Criteria	I1. The paper must be in the context of CE and Software Engineering.
	I2. The paper must report a primary or a secondary study.
	I3. The paper must provide data to answer at least one of the research questions.
	I4. The paper must be written in the English language.

Exclusion Criteria	E1. Duplicate publication/self-plagiarism.
	E2. Register of proceedings/posters.
	E3. Papers that are not peer-reviewed.
Technical Report	https://bit.ly/3DLj0uM

3.2. Execution

The search has been performed with Scopus by March 13th, 2021. It resulted in 1125 suggestions of articles, from which we selected 33 papers [S1][S2][S5][S14][S16][S17] [S18] [S20] [S22] [S25] [S26] [S28] [S29] [S31] [S42][S43][S46][S48][S49][S50][S51] [S53][S57][S59][S61][S62][S64][S68][S69][S70][S72][S76] because they provide models or experiment strategies of CE and deal with other contexts beyond the web context.

The papers were selected according to the defined inclusion and exclusion criteria. Then, two researchers analyzed the collection of selected sources to respond to each research question. The level of agreement was high among researchers, and the differences were resolved by analyzing the papers together. One last researcher reviewed the selection process—the snowballing identified 43 additional primary sources from 2007 until 2021. The final set of papers contains 76 selected papers. We consider this final set of articles relevant because of the different search strategies and objectives of our study and the appearance of [S6][S13][S15][S18][S24][S25][S28][S31][S42][S45][S46][S47][S48] [S49] [S50] [S51][S52][S53][S54][S55][S57][S58][S59][S65][S66] [S68] [S70] [S76], which are not present in the dataset of the last and large systematic literature review [Auer et al., 21].

4. Results

4.1. Continuous Experimentation Expressions and Definitions (RQ1)

In the selected papers, we identified 21 different expressions to describe the practice of applying experiments to guide a software system development process. Despite some similarities observed in their definition, some of these expressions have different definitions in various papers. Therefore, to facilitate the analysis, we decided to group the less cited expressions under the most cited ones. To do that, we first identified five terms that have citations in ten or more papers: Continuous Experimentation (30), Online Controlled Experiment (16), Data-Driven Development (12), Innovation Experiment System (10), and A/B Tests (10). However, analyzing the "Online Controlled Experiment" definitions, we realized that most of these matched this expression with A/B Tests. So, we decided to group the expression "A/B Tests" under "Online Controlled Experiment," reducing the categorization to four groups. The other terms were organized into these groups first by the similarity of their definitions in the papers: "Systematic Experimentation" and "Controlled Continuous Experimentation" into "Continuous Experimentation"; "Experiment-Driven Software Development," "Customer-Driven Development," and "Experiment-Driven Approach" into "Data-Driven Development"; and "Continuous Innovation," and "Innovation Process" into "Innovation Experiment System." Then, the remaining expressions were analyzed in its papers, and we identified that all of them were matched to A/B Tests. So, we organized

them into "Online Controlled Experiment." All the researchers agreed with this categorization. Table 2 shows all the found expressions and their categorization. In this table, the 'Qty' column indicates the hits of each term in the papers. All the cited expressions were counted when a paper had citations to more than one expression. Still, only one hit was calculated for each term in each paper.

Table 2. Used expressions to describe the practice of applying experiments to guide the development process of a software system.

Group expression	Expressions	Qty
Continuous Experimentation	Continuous Experimentation [S1][S2][S3][S13][S19][S20][S22] [S24][S25][S26][S27][S28][S29][S34][S35][S40][S44][S45][S46] [S52][S53][S62][S63][S64][S65][S68][S70][S73][S75][S76]	30
	Systematic Experimentation [S40]	1
	Controlled Continuous Experimentation [S69]	1
Online Controlled Experiment	Online Controlled Experiment [S10][S11][S13][S15][S16][S17] [S18][S31][S33][S37][S38][S39][S42][S43][S55][S59]	16
	A/B Tests [S10][S21][S30][S36][S37][S39][S43][S59][S72]	9
	Experimentation [S9][S15][S23][S32][S49][S50][S51][S74]	8
	Controlled Experiments [S30][S36][S72]	3
	Live Experimentation [S21][S67]	2
	Control/Treatment [S36]	1
	Parallel Flights [S36]	1
	Randomized Experiments [S36]	1
	Split Tests [S36]	1
	Test-and-learn [S55]	1
Data-Driven Development	Data-Driven Development [S5][S6][S14][S41][S43][S52][S55] [S56][S58][S59][S61][S71]	12
	Experiment-Driven Software Development [S43][S48][S53][S76]	4
	Customer-Driven Development [S57]	1
	Experiment-Driven Approach [S41]	1
Innovation Experiment System	Innovation Experiment System [S4][S6][S12][S34][S44][S55][S60] [S62][S66][S7]	10
	Continuous Innovation [S22][S34][S60][S66]	4
	Innovation Process [S8]	1

4.2. Continuous Experimentation Processes (RQ2)

Twenty-four models are proposed in the analyzed papers to guide the CE development process, implant CE into an organization, or deal with specific CE aspects [Table 3]. We classified them into six dimensions. First, development processes establish ordered activities to develop a software system guided by experimentation. Second, maturity processes propose a path to transition a traditional development process into an experimentation-driven one. Third, architecture models illustrate the software experiment structures. Fourth, logical flows show an experiment's paths. Finally, lifecycle and management help structure these specific parts of the experiment.

Each model was analyzed in its activities and purposes by two researchers. As a result, we classified eleven items as development processes, six as maturity processes, three as architecture models, two as logical flows, one as lifecycle, and one as management.

Table 3. CE models proposed in the technical literature.

Dimension	Model Name	Characteristics
Development Processes	Facebook's deployment pipeline [S21]	Development and deployment with canary release
	Hypotheses Engineering [S48]	Creating and managing hypothesis
	Explanatory CTP model for customer touchpoints and feedback data collection [S66]	Model focused on the interactions with customers
	HYPEX model [S56]	Shows how to close the "open-loop problem" between requirements and user data
	Unnamed model [S4] [S12]	CE model for embedded systems
	The HURRIER Process [S46]	CE model for business-to-business (B2B) systems
	The Qualitative/quantitative Customer-driven Development (QCD) model [S57] [S58]	Focused on customer feedback techniques to generate hypotheses
	Bing's experimentation process [S35]	Focused on validating data to iterate, ship, or abandon the hypothesis
	Experimentation Process Framework [S43]	Detailed CE model with activities, artifacts, inputs/outputs, and stored data
	Fagerholm et al. process [S19]	Based on the Lean Startup methodology, lists activities and roles
	RIGHT process model for Continuous Experimentation [S20]	Based on the Lean Startup methodology, lists activities and roles (update of [S19])
Maturity Processes	Transitioning towards experiment-driven development [S41]	Areas that the company needs to evolve up to CE
	Experimentation Evolution Model [S14]	Areas that the company needs to evolve up to CE
	Experimentation Growth (EG) Model [S15]	Areas that the company needs to evolve up to CE (update of [S14])
	Data-Driven Development Adoption Process [S59]	Steps that a company needs to follow to achieve CE
	eXperimentation Progression (XPro) model [S50]	Steps that a company needs to follow to achieve CE
	The Stairway to Heaven (StH) model [S6] [S34] [S60]	Steps that a company needs to follow to achieve CE
Architecture Models	Bing's experiment system architecture [S38]	Architecture for experimentation
	Giaimo and Berger model [S25]	Architecture for experimentation in automotive systems
	Evidence-Based Engineering [S5]	Architecture for experimentation in smart systems
Logical Flow	High-level flow for A/B test [S37] [S10]	Architecture for A/B Test
	Logic flow for A/B test [S38]	Architecture for A/B Test
Lifecycle	The experiment lifecycle [S17] [S18]	Experiment lifecycle
Management	A model of hypotheses engineering in startups [S52]	Management of hypothesis

4.3. Experiments Strategies (RQ3)

We identified several experimental strategies to adopt when performing CE. We perceive that A/B testing is the most applied and known experimental strategy cited in the selected studies. Our findings identified 47 different empirical strategies, including A/B tests. The experimental strategy tests a hypothesis and determines how the software will be updated. They determine how the experiment will be conducted, who will participate in it, in which project phase it will occur, what type of user data will be extracted, and how it will be analyzed.

Analyzing the experimental strategies, we perceived qualitative and quantitative approaches. Also, the strategies have different goals. For example, one experiment can use different strategies to test a hypothesis. An A/B Test can be deployed as a canary release and utilize overall evaluation criteria (OEC) to analyze the results for a quantitative example. Thus, we identified three main goals in the quantitative strategies [Table 4].

We named **Controlled experiments** (six items) the strategies that deal with the form of the experiments, i.e., how the experiments will be conducted. These strategies normally involve the end-user after developing the product or feature. Similarly, we called **Metrics measurement** (13 items) the strategies that indicate what type of user data will be extracted, which will also determine how this data will be analyzed. Some of these strategies require code parametrization in the software, while others can be measured externally. Finally, we named **Deploy mode** (nine items) the strategies that determine who will be selected for the experiment, influencing how the analysis will be conducted. These strategies are always applied in the deployment phase.

For qualitative strategies [Table 4], we identified two main groups with different goals, which we named **Participatory requirements** (six items) and **Partial appraisal** (13 items). These groups are other than the quantitative strategies because the participatory requirements strategies need the client or user representative to participate in the requirements elicitation phase. So, it defines who will participate in the experiment and the project phase. The experiments' conduction and the data collected and analyzed shall be selected from the partial appraisal strategies. However, these strategies can also be utilized without a participatory requirement, determining other selection of participants and different project phases. It can even be used with quantitative strategies to extract data that metrics measurement could not obtain.

Table 4. Experiment strategies categorization.

Quantitative strategies	**Controlled experiments**	A/B test, A/B/n test (or multivariate tests - MVT), Quasi-controlled experiments, MVP/MVF, Cross-over experimental design, and multi-armed bandits.
	Metrics measurement	Landing pages, fake door tests, wizard of oz MVP, metaheuristic search, bug reports, support logs, Google analytics, advertising, BASES testing, labs website, internal metrics collection, cognitive mapping, and overall evaluation criteria.
	Deploy mode	Alpha and beta testing (or early-access), canary releases (or partial rollouts or canary flying), blue/green deployment, gradual rollout, dark launches (or passive launch), parallel execution, serial execution, downsampled execution, selected customers (or proxy/lead users or expert reviews).
Qualitative strategies	**Participatory requirements**	Participatory design (or cooperative design, or joint-application design), scenarios, user stories, use cases, joint requirements sessions, and solution/innovation jams.
	Partial appraisal	Case studies (or field experiments or user studies), focus groups, surveys, interviews, observation, mockups (or sketches), prototypes, walkthroughs, feature voting, open testing, customer unit workshop (or customer conference or trade show testing), product seminar, and ethnographic studies.

5. Discussion and Implications

5.1. Continuous Experimentation Expressions and Definitions

We noticed that the expressions "Continuous Experimentation," "Data-Driven Development," and "Innovation Experiment System" are the ones having more

similarities to themes expressed in their definitions. The noticed themes were recurrence, a data-driven approach, and the use of user data to validate the experiments. It indicates that these expressions share the same intentions. However, the papers using the term "Innovation Experiment System" come mostly from the works of one research group [S4][S6][S7][S12][S34][S44][S55][S60][S66]. On the other hand, the expression "Online Controlled Experiment" diverges from the others because almost half the papers use this expression as synonymous with A/B Tests in the online domain. Besides, it has many citations to user data, but citations to data-driven approaches and recurrence are less common. A/B Tests can be used both in a data-driven and non-data-driven company. However, the three other expressions are applicable only in companies intending to be guided by the behavior of the end-users, transforming the possible requirements into hypotheses, and even creating theories through data obtained from the users.

This result expresses the current lack of consensus among researchers about which expressions should refer to the continuous experimentation approach in software development. The bigger quantity of CE usage indicates a trend to adopt this expression. However, the other terms are still very used. We believe that these expressions should be better distinguished to facilitate the research and express different approaches to software experimentation. Future works are needed to make clear this distinction.

Table 5. Phases of experimentation classify CE development processes. In each stage, "S" means "superficial," "D" means "detailed," and “NA” means “not addressed.”

Process Name	Experimentation Phases				
	Ideation	Experiment Design	Implementation	Execution	Analysis
Facebook's deployment pipeline [Feitelson et al., 13]	NA	NA	S	D	NA
Hypotheses Engineering [Melegati et al., 19a]	D	S	NA	S	S
Explanatory CTP model for customer touchpoints and feedback data collection [Sauvola et al., 15]	D	D	NA	NA	D
HYPEX model [Olsson and Bosch, 14]	S	S	S	S	D
Unnamed model [Eklund and Bosch, 12] [Bosch and Eklund, 12]	S	NA	S	D	S
The HURRIER Process [Mattos et al., 20b]	D	NA	D	D	NA
The Qualitative/quantitative Customer-driven Development (QCD) model [Olsson and Bosch, 15a] [Olsson and Bosch, 15b]	D	D	NA	S	D
Bing's experimentation process [Kevic et al., 17]	S	S	S	S	D
Experimentation Process Framework [Mattos et al., 18a]	D	S	D	S	D
Fagerholm et al. process [Fagerholm et al., 14]	S	S	D	S	D
RIGHT process model for Continuous Experimentation [Fagerholm et al., 17]	S	S	D	S	D

5.2. Continuous Experimentation Processes

This study considered only the eleven development processes identified in [Table 3] because its objective is generally to analyze CE's conduction. Two of them were designed for specific software contexts [Eklund and Bosch 2012] [Bosch and Eklund 2012] [Mattos et al. 2020b], and the others are for general purposes. Table 5 shows

these eleven processes classified according to the five phases of experimentation presented by [Auer et al. 2021], identifying whenever a phase is cited superficially, detailed, or not addressed in the model. We consider that a stage is detailed if more than one step in that phase or if the one-step has associated characteristics. This classification shows that none of the processes has all phases detailed. Six processes (54,5%) do not present all the stages. However, all the processes have at least one detailed phase. It indicates that all these processes aim to explore a specific part of the development process guided by experimentation. Since the processes have different clear stages, we understand that the processes have complementary parts. Therefore, the practitioners could benefit not from choosing just one of them but from utilizing some of them together.

5.3. Continuous Experimentation Strategies

The findings indicate a lack of consensus about which strategies to use for a CE process, even with many references about A/B Tests, mostly through collecting metrics. Therefore, more studies are needed to show which strategies should be used in CE processes. Furthermore, it is important to understand the qualitative strategies and their role in CE, particularly when quantitative data does not support answering questions or refuting hypotheses. For instance, it does not indicate which part of the software needs improvement. Qualitative strategies are required in this regard [Ros 2020].

Furthermore, there is a need to discuss whether participatory requirements should be used within the CE process, as stated in some analyzed papers, or should not, as in the processes in the matrix. For example, the works [Melegati et al. 2019a] [Melegati et al. 2019b], and [Melegati et al. 2020c] advocate that the requirement concept is inadequate in the CE context and that the *hypotheses engineering* should replace the requirements engineering in these cases. According to them, requirements engineering is an important component of "traditional requirements-driven development." However, in "experiment-driven software development," the *hypotheses* guide the elicitation of the users' needs, and they evolve together with the coding. So, as experiment-driven software development, CE would benefit from *hypotheses engineering* to better identify, prioritize, specify, analyze, and manage its hypotheses and reduce the waste of resources and time. In this way, the participatory requirements strategies should be adapted to this new concept to gain statistical relevance and be used as useful data-driven strategies in CE processes in software engineering.

6. Threats to Validity

As expected in any empirical study, threats to validity deserve attention. First, regarding reliability, we selected the works from technical literature following good search practices and considering an initial set of secondary studies (Section 2). Although they do not assure full replication, we described the elementary features that allow repeatability of results. Further, we have provided each article's inclusion and exclusion criteria [Table 1] and applied snowballing techniques to enlarge coverage and reduce this threat, including six secondary studies. Also, we have a larger list of selected primary sources than most previously identified secondary studies, including papers that do not appear.

To mitigate the bias of the selected papers and the interpretation bias of researchers, two researchers determined the articles, and two others reviewed the final

set. The research protocol aims at promoting its data traceability. Additionally, an *ad-hoc* analysis supported this literature study. Thus, the interpretation and synthesis of the articles can be subjective. However, the researchers' experience with coding practices and data synthesis can naturally influence how such an analysis is conducted.

7. Conclusion

Continuous Experimentation has become widely known as a valuable development practice by practitioners and researchers. However, understanding the planning and implementation of CE in SE is still difficult because of the plurality of interpretations in the technical literature. In this context, based on the current body of knowledge examined utilizing a literature study, we characterize CE in its definitions, processes, and experimentation strategies.

We identified results regarding the definitions in which "continuous experimentation" shares the same intentions as "data-driven development" and "innovation experiment system," However, the distinct approaches are different from "online controlled experiment," which is an expression for A/B Test. The lack of a common terminology creates difficulties for researchers to discover and understand the different terms used in the available studies. To the best of our knowledge, no other research has discussed these expressions' differences.

We also identified 24 models regarding CE, eleven of which were development processes. We noticed that these development processes share common activities. Still, each has parts that deal with different experimentation aspects, making them complementary. This plurality of expressions and diverse highlights makes selecting the appropriate process for a specific context a challenge.

Finally, we identified 47 experimentation strategies, categorized them into two groups, subcategorized them into five subgroups, and created a correlation matrix of the processes and the strategies. We noticed that the A/B test is the most applied strategy known by both practitioners and researchers. However, the number of strategies found and the fact that most of them appear in few papers state that more studies are needed to determine the contributions of each strategy for CE.

We also identified that dealing with hypotheses is a little-explored challenge for CE. The relation between the conjectured software properties and requirements is unclear. Therefore, the recently raised concept of Hypotheses Engineering could help to align these approaches in the context of CE in SE. Continuous experimentation is not just collecting data, but it represents a systematic method having its concepts, processes, and strategies. It needs to be understood and aligned with the organization's strategic objectives to succeed in engineering software systems using CE. Therefore, despite its increasing use, it is impossible to observe a common terminology yet to support its characterization and use in SE. Further studies are necessary to organize such concepts and taxonomically represent them to make continuous experimentation less blurred in software systems projects.

Acknowledgments

This work is partially supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 and by CNPq. Prof. Travassos is a CNPq researcher (Grant 304234/2018-4) and CNE FAPERJ (Grant E-26/201.170/2021).

References

Basili, V. R., Caldiera, G., Rombach, H. D. (1994) "The goal question metric approach"; Encyclopedia of software engineering, 528-532.

Kuhrmann, M., Fernández, D. M., Daneva, M. (2017) “On the pragmatic design of literature studies in software engineering: an experience-based guideline”; ESE 22.6.

Wohlin, C. (2014) "Guidelines for snowballing in systematic literature studies and a replication in software engineering"; Proceedings of the 18th EASE.

Selected Publications

[S1] Auer, F. and Felderer, M. (2018) "Current state of research on continuous experimentation: a systematic mapping study"; 2018 44th Euromicro SEAA, IEEE.

[S2] Auer, F., Lee C. S., and Felderer, M. (2020) "Continuous experiment definition characteristics"; 2020 46th Euromicro SEAA, IEEE.

[S3] Auer, F. et al. (2021) "Controlled experimentation in continuous experimentation: Knowledge and challenges"; Information and Software Technology 134, 106551.

[S4] Bosch, J., Eklund, U. (2012) "Eternal embedded software: Towards innovation experiment systems"; ISoLA, Springer, Berlin, Heidelberg.

[S5] Bosch, J. and Olsson, H. H. (2016) "Data-driven continuous evolution of smart systems"; 2016 IEEE/ACM 11th SEAMS, IEEE.

[S6] Bosch, J., and Olsson, H. H. (2017) "Toward evidence-based organizations: lessons from embedded systems, online games, and the Internet of Things"; IEEE Software 34.5.

[S7] Bosch, J. (2012) "Building products as innovation experiment systems"; International Conference of Software Business, Springer, Berlin, Heidelberg.

[S8] Bosch-Sijtsema, P. and Bosch, J. (2015) "User involvement throughout the innovation process in high-tech industries"; JPIM 32.5, p. 793-807.

[S9] Buchert, T. et al. (2015) "A survey of general-purpose experiment management tools for distributed systems"; Future Generation Computer Systems 45, p. 1-12.

[S10] Crook, T., et al. (2009) "Seven pitfalls to avoid when running controlled experiments on the web"; 15th ACM SIGKDD.

[S11] Deng, A., and Shi, X. (2016) "Data-driven metric development for online controlled experiments: Seven lessons"; 22nd ACM SIGKDD.

[S12] Eklund, U. and Bosch, J. (2012) "Architecture for large-scale innovation experiment systems"; IEEE/IFIP Conference and European Conference, IEEE.

[S13] Esteller-Cucala, M., Fernandez, V., and Villuendas, D. (2020) "Towards data-driven culture in a Spanish automobile manufacturer: A case study"; JIEM 13.2.

[S14] Fabijan, A., et al. (2017) "The evolution of continuous experimentation in software product development: from data to a data-driven organization at scale"; IEEE/ACM 39th ICSE, IEEE.

[S15] Fabijan, A., et al. (2018a) "Experimentation growth: Evolving trustworthy A/B testing capabilities in online software companies"; J. Softw.: Evol. Process 30.12.

[S16] Fabijan, A. et al. (2018b) "Online controlled experimentation at scale: an empirical survey on the current state of A/B testing"; 44th Euromicro SEAA, IEEE.

[S17] Fabijan, A. et al. (2018c) "The online controlled experiment lifecycle"; IEEE Software 37.2, p. 60-67.

[S18] Fabijan, A., et al. (2019) "Three key checklists and remedies for trustworthy analysis of online controlled experiments at scale"; IEEE/ACM ICSE-SEIP, IEEE.

[S19] Fagerholm, F. et al. (2014) "Building blocks for continuous experimentation"; 1st international workshop on rapid continuous software engineering.

[S20] Fagerholm, F., et al. (2017) "The RIGHT model for continuous experimentation"; Journal of Systems and Software 123, p. 292-305.

[S21] Feitelson, D. G., Frachtenberg, E. and Beck, K. L. (2013) "Development and deployment at Facebook"; IEEE Internet Computing 17.4, p. 8-17.

[S22] Fitzgerald, B. and Stol, K.J. (2017) "Continuous software engineering: A roadmap and agenda"; Journal of Systems and Software 123, p. 176-189.

[S23] Gerostathopoulos, I., et al. (2018) "Cost-aware stage-based experimentation: challenges and emerging results"; 2018 IEEE ICSA-C, IEEE.

[S24] Giaimo, F. and Berger, C. (2017) "Design criteria to architect continuous experimentation for self-driving vehicles"; IEEE ICSA.

[S25] Giaimo, F. and Berger, C. (2020) "Continuous Experimentation for Automotive Software on the Example of a Heavy Commercial Vehicle in Daily Operation"; ECSA.

[S26] Giaimo, F., et al. (2016) "Continuous experimentation on cyber-physical systems: challenges and opportunities"; Proceedings of the scientific workshop XP2016.

[S27] Giaimo, F., Berger, C., and Kirchner, C. (2017) "Considerations about continuous experimentation for resource-constrained platforms in self-driving vehicles"; ECSA.

[S28] Giaimo, F., Andrade, H. and Berger, C. (2019) "The automotive take on continuous experimentation: a multiple case study"; 45th SEAA, IEEE.

[S29] Giaimo, F., Andrade, H. and Berger, C. (2020) "Continuous experimentation and the cyber-physical systems challenge: An overview of the literature and the industrial perspective"; Journal of Systems and Software 170, 110781.

[S30] Gomez-Uribe, C. A. and Hunt, N. (2015) "The Netflix recommender system: Algorithms, business value, and innovation"; ACM TMIS 6.4, p. 1-19.

[S31] Gupta, S. et al. (2019) "Top challenges from the first practical online controlled experiments summit"; ACM SIGKDD Explorations Newsletter 21.1, p. 20-35.

[S32] Gutbrod, M., Münch, J. and Tichy, M. (2019) "How do software startups approach experimentation? Empirical results from a qualitative interview study"; PROFES.

[S33] Jiang, S., Martin, J. and Wilson, C. (2019) "Who's the Guinea Pig? Investigating Online A/B/n Tests in-the-Wild"; Proceedings of the ACM FAccT.

[S34] Karvonen, T. et al. (2015) "Hitting the Target: Practices and Steps for Moving Towards Innovation Experiment Systems"; LNBIP.

[S35] Kevic, K. et al. (2017) "Characterizing experimentation in continuous deployment: a case study on bing"; 2017 IEEE/ACM 39th ICSE-SEIP, IEEE.

[S36] Kohavi, R., Henne, R. M. and Sommerfield, D. (2007) "Practical guide to controlled experiments on the web: listen to your customers, not to the hippo"; 13th ACM SIGKDD.

[S37] Kohavi, R., et al. (2012) "Trustworthy online controlled experiments: Five puzzling outcomes explained"; 18th ACM SIGKDD.

[S38] Kohavi, R., et al. (2013) "Online controlled experiments at large scale"; 19th ACM SIGKDD.

[S39] Kohavi, R., et al. (2014) "Seven rules of thumb for website experimenters"; 20th ACM SIGKDD.

[S40] Lindgren, E. and Münch, J. (2015) "Software development as an experiment system: A qualitative survey on the state of the practice"; ICASD.

[S41] Lindgren, E. and Münch, J. (2016) "Raising the odds of success: the current state of experimentation in product development"; IST 77, p. 80-91.

[S42] Liu, S. et al. (2019) "Enterprise-Level Controlled Experiments at Scale: Challenges and Solutions"; 45th Euromicro Conference on SEAA, IEEE.

[S43] Mattos, D. I., et al. (2018a) "An activity and metric model for online controlled experiments"; PROFES, Springer, Cham.

[S44] Mattos, D. I., Bosch, J., and Olsson, H. H. (2018b) "Challenges and strategies for undertaking continuous experimentation to embedded systems: Industry and research perspectives"; ICASD.

[S45] Mattos, D. I., et al. (2020a) "Automotive a/b testing: Challenges and lessons learned from practice"; 46th Euromicro Conference on SEAA, IEEE.

[S46] Mattos, D. I., et al. (2020b) "Experimentation for business-to-business mission-critical systems: a case study"; Proceedings of the ICSSP.

[S47] Melegati, J. and Wang, X. (2020) "Hypotheses Elicitation in Early-Stage Software Startups Based on Cognitive Mapping"; ICASD, Springer, Cham.

[S48] Melegati, J., Wang, X., and Abrahamsson, P. (2019a) "Hypotheses Engineering: first essential steps of experiment-driven software development"; 4th International Workshop on RCoSE/DDrEE, IEEE.

[S49] Melegati, J., et al. (2019b) "Enablers and inhibitors of experimentation in early-stage software startups"; PROFES, Springer, Cham.

[S50] Melegati, J., Edison, H., and Wang, X. (2020a) "XPro: a Model to Explain the Limited Adoption and Implementation of Experimentation in Software Startups"; IEEE TOSEM.

[S51] Melegati, J., et al. (2020b) "MVP and experimentation in software startups: a qualitative survey"; 46th Euromicro Conference on SEAA, IEEE.

[S52] Melegati, J., Guerra, E., and Wang, X. (2020c) "Understanding Hypotheses Engineering in Software Startups through a Gray Literature Review"; IST, 106465.

[S53] Melegati, J. (2019) "Improving requirements engineering practices to support experimentation in software startups"; 27th ACM Joint Meeting on ESEC/FSE.

[S54] Olsson, H. H. and Bosch, J. (2013a) "Post-deployment data collection in software-intensive embedded products"; ICSOB, Springer, Berlin, Heidelberg.

[S55] Olsson, H. H. and Bosch, J. (2013b) "Towards data-driven product development: A multiple case study on post-deployment data usage in software-intensive embedded systems"; LESS, Springer, Berlin, Heidelberg.

[S56] Olsson, H. H. and Bosch, J. (2014) "From opinions to data-driven software R&D: A multi-case study on how to close the 'open loop' problem"; 40th EUROMICRO SEAA.

[S57] Olsson, H. H. and Bosch, J. (2015a) "Towards continuous validation of customer value"; Scientific Workshop Proceedings of the XP2015.

[S58] Olsson, H. H. and Bosch, J. (2015b) "Towards continuous customer validation: A conceptual model for combining qualitative customer feedback with quantitative customer observation"; International Conference of Software Business. Springer, Cham.

[S59] Olsson, H. H. and Bosch, J. (2019) "Data-driven development: Challenges in online, embedded and on-premise software"; PROFES, Springer, Cham.

[S60] Olsson, H. H., Bosch, J. and Alahyari, H. (2013) "Towards R&D as innovation experiment systems: A framework for moving beyond agile software development"; IASTED ().

[S61] Olsson, H. H., Bosch, J. and Fabijan, A. (2017) "Experimentation that matters: a multi-case study on the challenges with A/B testing"; ICSOB, Springer, Cham.

[S62] Rissanen, O. and Münch, J. (2015) "Continuous experimentation in the B2B domain: a case study"; IEEE/ACM 2nd International Workshop on RCoSE, IEEE.

[S63] Ros, R. and Bjarnason, E. (2018) "Continuous experimentation scenarios: a case study in e-Commerce"; 44th Euromicro Conference on SEAA, IEEE.

[S64] Ros, R. and Runeson, P. (2018) "Continuous experimentation and a/b testing: A mapping study"; 2018 IEEE/ACM 4th International Workshop on (RCoSE), IEEE.

[S65] Ros, R. (2020) "Continuous Experimentation with Product-Led Business Models: A Comparative Case Study"; ICSOB, Springer, Cham.

[S66] Sauvola, T. et al. (2015) "Towards customer-centric software development: a multiple-case study"; 41st Euromicro Conference on SEAA, IEEE.

[S67] Schermann, G. et al. (2016) "Bifrost: Supporting continuous deployment with automated enactment of multi-phase live testing strategies"; 17th IMC.

[S68] Schermann, G., Cito, J., and Leitner, P. (2018a) "Continuous experimentation: challenges, implementation techniques, and current research"; Ieee Software 35.2, p. 26-31.

[S69] Schermann, G., et al. (2018b) "We're doing it live: A multi-method empirical study on continuous experimentation"; Information and Software Technology, v. 99.

[S70] Sveningson, R., Mattos, D. I. and Bosch, J. (2019) "Continuous experimentation for software organizations with low control of roadmap and a large distance to users: An exploratory case study"; PROFES, Springer, Cham.

[S71] Tang, D., et al. (2010) "Overlapping experiment infrastructure: More, better, faster experimentation"; 16th ACM SIGKDD International Conference on KDD.

[S72] Xu, Y. and Chen, N. (2016) "Evaluating mobile apps with A/B and quasi A/B tests"; 22nd ACM SIGKDD International Conference on KDD.

[S73] Yaman, S. G., et al. (2016) "Transitioning towards continuous experimentation in a large software product and service development organization–a case study"; PROFES.

[S74] Yaman, S. et al. (2017) "Notifying and involving users in experimentation: ethical perceptions of software practitioners"; ESEM.

[S75] Yaman, S., Mikkonen, T., and Suomela, R. (2018) "Continuous experimentation in mobile game development"; 2018 44th Euromicro Conference on SEAA, IEEE.

[S76] Yaman, S., et al. (2020) "Patterns of user involvement in experiment-driven software development"; Information and Software Technology, v. 120, p. 106244.