Patterns to Support a Continuous Experimentation Process

ALEXANDRE BITTENCOURT FARIA, Centro Paula Souza - FATEC Zona Sul, Brazil
DANILO ALVES DE JESUS, National Institute for Space Research, Brazil
FERNANDO PEREIRA, National Center for Monitoring and Early Warnings of Natural Disasters, Brazil
JOELMA CHOMA, National Institute for Space Research, Brazil
LUIS RICARDO ARANTES FILHO, National Institute for Space Research, Brazil
VANESSA GOMES ALBUQUERQUE, Group Being Educational - University Guarulhos - UNIVERITAS, Brazil
EDUARDO MARTINS GUERRA, National Institute for Space Research, Brazil

The demand for software application development has grown exponentially in recent years. The development of applications that better adjust to the real needs of the users is a challenge. Understanding the behavior of users to improve certain points of the software is also not an easy task. Often, this assessment is done by assumptions rather than being based on real data. One of the approaches to understanding user behavior for software application development is through continuous experimentation. Continuous experimentation aims to obtain information about the behavior and preferences of the users directly or indirectly through the analysis of the data generated by multiple experiments. A continuous experimentation process can generate a large amount of data over time. In this paper, we present a set of patterns related to the use continuous experimentation that aims the continuous improvement of software focused on the correct interpretation of data and the most appropriate way to systematize the experimentation process of software products and services.

Categories and Subject Descriptors: D.2.8 [**Software and its engineering**]: Software creation and management—*Metrics*

General Terms: Continuous Experimentation

Additional Key Words and Phrases: Continuous Experimentation, Patterns, Experiment-driven Software Development

ACM Reference Format:

Faria, A., Jesus, D., Pereira, F., Choma, J., Arantes Filho, L. R., Albuquerque, V., Guerra E. M. 2018. Patterns to Support a Continuous Experimentation Process. HILLSIDE Proc. of Conf. on Pattern Lang. of Prog. V (October 2018), 15 pages.

1. INTRODUCTION

According to [Kuhn 1981], an experiment is a type of scientific research in which the researcher manipulates and interpret independent variables and observes the variation in the dependent variables concomitantly with the manipulation of the independent variables.

Author's address: Faria, A. B. Centro Paula Souza - FATEC Zona Sul, Brazil, email: abittencourtfaria@gmail.com; Jesus, D. A. National Institute for Space Research, Brazil, email: daniloa47@gmail.com; Pereira, F. National Center for Monitoring and Early Warnings of Natural Disasters, Brazil, email: fernando.opc@gmail.com; Choma, J. National Institute for Space Research, Brazil, email: jh.choma@hotmail.com; Arantes Filho, L. R. National Institute for Space Research, Brazil, email: luisricardoengcomp@gmail.com; Albuquerque, V. Group Being Educational - University Guarulhos - UNIVERITAS, Brazil, email: vanessa.ga@gmail.com; Guerra, E. M. National Institute for Space Research, Brazil, email: guerraem@gmail.com

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission. A preliminary version of this paper was presented in a writers' workshop at the 12th Latin American Conference on Pattern Languages of Programs (SLPLoP). SLPLoP'18, NOVEMBER 20-23, Valparaíso, Chile. Copyright 2018 is held by the author(s). HILLSIDE 978-1-941652-11-4.

The majority of the information about new software technologies (processes, methods, techniques, and tools) is still based on advertisements or self-opinions. However, scientific research cannot be based on personal interest or opinions [Juristo and Moreno 2010]. In this sense, experimentation provides a systematic, disciplined and controlled path to evaluate human processes and activities [Wohlin et al. 2012].

Experiment-driven development approaches contribute to providing justifications for the use or not of technologies, based on indications about their effectiveness for software quality improvement. Thus, the results of experimental studies performed in different research scenarios can be used as starting points to define a set of criteria that support the decision-making process regarding the use of software technologies. One way to reduce the risks of software development is through continuous experimentation [Yaman et al. 2017].

Continuous experimentation (CE) is a development approach driven by experiments in products and services, which by iterative testing their assumptions can achieve success with less risk in an organization. The data collected through CE is the requirement that must support the development and continuous improvement cycles of these products and services, also allowing a fast decision-making process [Schermann 2017].

Recently, studies in the field of CE have pointed out that companies are increasingly interested in adopting this approach into their development process [Kevic et al. 2017] [Yaman et al. 2017] [Lindgren and Münch 2016]. By using CE, some of these companies succeeded in reducing development efforts and to better understand the needs of their customers. Introducing CE required organizational commitment and development of skills for experimentation. However, [Yaman et al. 2017] claimed that there are still few studies about how introducing CE into an organization with an already established development process.

To help fill this gap, we sought to identify best practices within this research segment based on existing case studies in the literature. The objective of our study was a better understanding of the process of introducing and applying the stages of CE already established in an organization.

In this paper, we present a set of patterns to promote CE to improve software products and services. We have elaborated our patterns from related studies that indicate criteria for success and the construction of fundamental experimentation skills to systematize and support patterns based-on experiments. The proposed patterns present possible solutions that fit into several situations, such as the correct definition of hypotheses; the choice of the right people for experimentation; how to train and guide people through the process of experimentation; how to optimize the experimentation process and, how to interpret correctly the results of the experiments, avoiding unsubstantiated assumptions.

The integration of patterns based on CE aims a better understanding, perception, motivation, and identification with their respective tasks of software engineers, data scientists, systems analysts, programmers, testers and all professionals involved in the software development process. Experimentation is a concept that must capture what people feel about the products, systems, and services to improve the user experience and consequently the success of projects and collaboration in organizations. It is worth noting that the proposed patterns can be applied together, seeking the continuous improvement of the experimentation process.

This paper is arranged as follows. Section 2 describes the concepts and examples of CE. Section 3 describes the methodology in the development of the patterns. Section 4 introduces the patterns for CE, an overview of each one and the relationships between them. Sections 5,6,7,8 and 9 explain in detail the proposed patterns indicating their motivations, positive and negative points, forces and main features. Finally, Section 10 refers to the discussions and concluding remarks.

2. BACKGROUND

[Fagerholm et al. 2014] have suggested CE as an approach for promoting continuous improvement of software products and services in line with customers' needs. In order to understand the main points of product improvement, they have emphasized that the need to obtain direct or indirect feedback from the customers and then implement each point observed by them. However, obtaining accurate feedback may not be easy and its response cycle can be very slow due to the lack of efficient mechanisms of data collection and analysis [Olsson and Bosch 2014b].

In their work, [Fagerholm et al. 2014] proposed a model to link experimentation on the product and technical level to the product vision and strategy on the business level. This model allows focused testing of business hypotheses[1] and assumptions[2], which can be turned into faster decision-making and reaction to customer needs.

According to [Olsson and Bosch 2014a], software-intensive systems companies need to continually evolve their practices. They refer to “stairway to heaven” pattern to discuss the typical evolution path for companies moving towards continuous deployment of software.

Studies of web-controlled experiments establishing a relationship between changes and influence of user behavior were addressed by [Kohavi et al. 2009]. Their study indicated significant learning and return-on-investment (ROI) when development teams listen to their customers. Web-oriented businesses use controlled experiments online to guide product development and accelerate innovative products and services. For example, Microsoft’s Bing has grown exponentially over time performing large-scale experiments. According to [Tang et al. 2010], at Google experimentation, is practically a mantra. The authors believe that the Google’ experimentation process can be generalized and applied by anyone interested in using experimentation to improve the search engines, for example.

[Kohavi et al. 2013] described multiple challenges related to organizational culture, engineering, and reliability in this company concerning their experimentation process. However, they highlighted that adoption of continuous experimentation has accelerated innovation and increased the annual revenue by hundreds of millions of dollars, allowing them to focus on ideas emerged from experiments. [Kohavi et al. 2014] shared seven practical rules for experimenters in order to help them optimize sites and provide new search challenges about applicability, exceptions, and extensions by adopting experimental criteria.

[Rissanen and Münch 2015] analyzed the challenges, benefits and organizational aspects of CE in the B2B domain. The results suggest that technical challenges are only part of the challenges a company faces in this transition. The company also needs to address customer and organizational culture challenges. Unique properties in each client’s business play an important role and need to be considered when designing experiments. In addition, the speed at which experiments can be conducted is relative to the speed at which production deployments can be made.

[Bosch and Eklund 2012] pointed out architectural challenges for CE in long-life embedded systems such as the ability to evolve and conduct experiments on the product already in use in a secure and controlled manner. In their work, they concluded that not all embedded systems are suitable for CE. [Giaimo and Berger 2017] proposed a set of criteria for experimentation in autonomous vehicles due to the critical aspects of security, real-time response and limited resources present in this type of system. Also, [Issa Mattos et al. 2018] pointed out challenges and strategies for undertaking CE in software companies that develop embedded systems. They identified twelve challenges divided into technical, business, and organizational areas; and strategies grouped into the architecture, data handling, and development processes categories.

[Kevic et al. 2017] suggested that a software process can be improved through continuous large-scale experimentation by evaluating small versions of software in a short time for end users. They proposed a framework for CE to support the process of hypothesis definition and the creation of a set of metrics to see how software modifications impact the client. This process was applied in the Bing search tool where about 21220 experiments were used from 2014 to 2017. Applying CE on a large scale can be a tricky task, due to a large number of experiments. For that reason, it is necessary to establish correctly the main elements that the process of experimentation should have. [Fabijan et al. 2017] proposed a model for the evolution of experimentation that provides guidance to practitioners on how to develop and scale CE process. Their model has three phases of evolution: technical, organizational and business evolution.

[1] Hypothesis is a proposed testable explanation for a phenomenon [Munezero et al. 2017].
[2] Assumptions refer to the aspects of your idea that is accepted as true or as certain to happen, without proof [Munezero et al. 2017].

[Fagerholm et al. 2017] presented the “Rapid Iterative Value Creation” model for CE. According them, the model describes the experimentation process, in which assumptions for product and business development are derived from the business strategy, systematically tested, and the results used to inform further development of the strategy and product. The main challenges this approach are proper, rapid design of experiments, advanced instrumentation of software to collect, analyze, and store relevant data, and the integration of experiment results in both the product development cycle and the software development process.

3. RESEARCH METHOD

The patterns to support a CE process were elaborated in three steps. At the first step, we carried out a literature review where the focus was mainly on works describing concrete examples of the use of the methodology in different contexts, especially the results and learning achieved involving developers and system users.

In the next step, all authors participated in brainstorming sessions to discuss the results of relevant studies identified in the previous step. The main goal was to look for and recognize directives and behaviors transverse to all the works. From this discussion, it was possible to notice the main problems faced and the practices to be adopted. Then, we identified the patterns described in this paper.

Finally, to refine the identified patterns, we undertaken a writers workshop using the “Fly on the wall pattern” [Lucrédio et al. 2004]. During this meeting, each author made a brief summary of the pattern written by him and then, listened in silence to the other authors discuss positive and improvement points of the pattern. Thereafter, the collection of comments and suggestions was used to refine the writing of the patterns.

4. PATTERNS TO SUPPORT A CE PROCESS

The patterns presented in this paper are intended to assist software practitioner in the configuration of steps to perform CE on software products and services and demonstrate ways to facilitate the process to establish hypotheses and to test them before delivering software features. These patterns are as recipes that fit the needs of a developer at a time of planning, execute and analyze the CE process. Figure 1 shows an overview of patterns and the relationship among them. As shown in Figure 1, the proposed patterns are contextualized as follows:

—**Pilote Experiments** pattern refers to start the CE process with simple experiments in order disseminate the experimentation culture and involve team members in this process.

—**Data Scientists Skills** pattern highlights the importance of data scientist role within the CE process in order to obtain useful and accurate information from experiments data.

—**Choose the Right Participants** pattern addresses the need to select experts to work on projects that contemplate CE. It is not trivial, but it is necessary to generate good results. The wrong choice of these experts can lead the project to failure.

—**Most Suitable Way** pattern refers to identify the most appropriate way to obtain results for better decision making. The results of the experiments allow the verification and analysis of related criteria, allowing process specialists a better analysis of their decisions based on predefined experimentation criteria.

—**Experimentation Knowledge Base** pattern suggests that instances of the continuous experimentation process can be stored for future queries and replication of new experiments.

With respect to the order of application, note that these patterns do not necessarily have to be applied in sequence, or in the same order as presented in this paper. Next, we describe the patterns in more detail.

5. PILOT EXPERIMENTS

5.1 Motivational Example

Many professionals face difficulties in introducing new concepts and methodologies in their daily lives. Most of the time, these difficulties arise from a high learning curve, by the lack of interest of the team or by misaligned

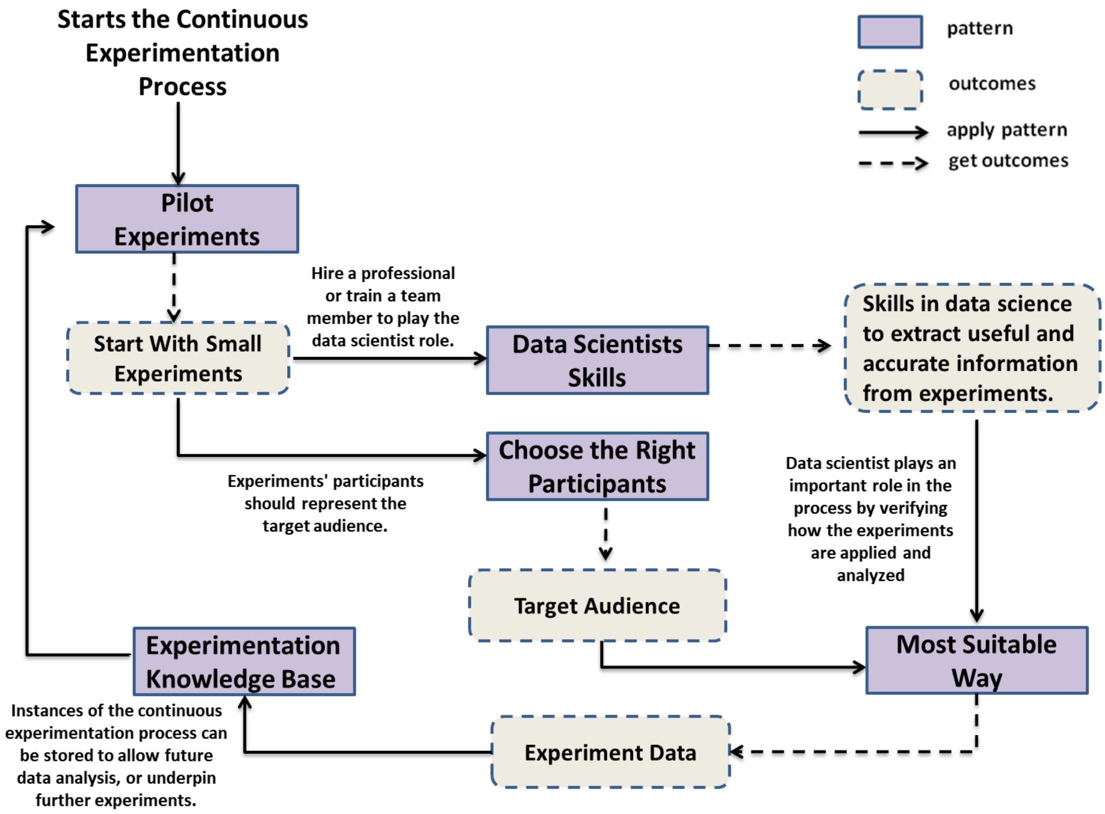

Fig. 1. Overview of the patterns and their relationships.

expectations in relation to the improvements that are intended to achieve. This is not different in the CE area, and deploying it in the work environment can generate frustrations if it is not seen as a change of behavior. Experimentation requires that the team be guided by data rather than by assumptions and opinions. It is a cyclical process and companies that have been successful in its use make them evolve from simple experiments in order to spread the culture of experimentation and then evolve them.

5.2 Context

For teams unfamiliar with the practice of CE, it may be difficult at the beginning of the methodology to formulate good hypotheses, build experiments, choose testers, and efficiently collect and analyze data, which makes it difficult to make assertive decisions.

5.3 Problem

How can we introduce the process of continuous experimentation with teams whose members do not know its benefits?

5.4 Forces

—Attempting to initiate the use of CE in a team can be difficult because of the lack of knowledge in the new methodology.

—If the team cannot visualize a basic example of an experiment in a practical way in their working context, the experimentation methodology is less likely to be implemented on a time and cost basis.

5.5 Solution

Choose small teams, create simple experiments that aim to test small parts of the product. These experiments should serve as trials to disseminate the experimentation culture serving as training, generating interest and better adaptation to the new methodology.

Choosing teams, for example, up to 5 members will favor training, once the experimenter will have a greater control. Create simple experiments, as for example, logs generators or messages customization of the applications, it will favor the experimentation once the experimenter can focus in the diffusion of methodology concepts and not in the in business rules. The same idea holds true for testing small parts of the product, which simplifies the initiation in the use of the CE.

5.6 Consequences

(+) Training the team with simple experiments from your work context will stimulate it and it will be easier to evolve into sophisticated experiments.

(+) Teams trained from simple experiments come to better understand how continuous experimentation works and can see in methodology a way of being guided by data rather than by assumptions.

(-) There may be resistance on the part of managers to allocate resources for the adoption of the methodology of continuous experimentation, since the methodology may not bring immediate results.

5.7 Known Uses

As an example, [Munezero et al. 2017] cited the case of Ericson company that when starting the use of CE sought to work from simplified experiments, with small teams and smaller modules of the product, training the team and then evolve their experiments in order to generate data and subsidize the decision making. Another example is the case of Solita, a Finnish company specialized in helping to digitize companies and services, where the developers implemented minimum viable feature (MVF) and instrumented for data collection, gradually refining the collection in order to train the work in the CE context.

6. DATA SCIENTISTS SKILLS

6.1 Motivational Example

Extracting knowledge from results generated from a CE process is not trivial. A lot of data usually are collected supposing that their analysis will lead to something useful. To analyze and interpret a large volume of data can require skills of a data scientist. Data scientists work with data to solve problems. Their work extends all the way from the collection of data to obtaining insights from the data captured. Thus, data scientist have skills in data collection, processing, manipulation, and interpretation using tools and techniques from mathematics, statistics, and computation.

6.2 Context

Continuous experimentation typically generate large amounts of data, which can be analyzed using data science tools and techniques (e.g., data mining techniques, machine learning, and data visualization). Skills in data science, allow software professionals to gain deeper understanding of the factors influencing the issues of interest, contributing to continuous improvement of product and process. Moreover, organizations that embrace experimentation can evolve their systems with automated optimization and real-time analyses.

6.3 Problem

How can we explore data scientist skills to extract useful and accurate knowledge from the results obtained through continuous experimentation?

6.4 Forces

—Experiments results can provide valuable information, but the volume of data to extract this information can be large and difficult to handle.
—There are many tools for extracting knowledge from large volumes of data, but team members are not able to interpret the results.
—Data scientist are increasingly important within CE process, but the team may not have enough resources to hire an expert in this area.
—The team members can develop some skills seeking knowledge in the area of data science, but they may not have time to explore all the resources available in this area.

6.5 Solution

Hire a professional or train a team member to play the data scientist role with skills to extract useful information from the data generated from continuous experimentation.

Logical thinking is an important skill to data analysis. However, programming knowledge enables the data scientist put their creativity into practice and extract data from answers to questions that have not yet been asked. Mathematics, statistic, and machine learning are fundamental in data science, mainly to manipulate a large volume of data. Moreover, at various stages of the data analysis process, interactions with databases will be required. Thus, it is very important the knowledge about different kind of databases and languages for querying.

6.6 Consequences

(+) Historical data to obtain better results in the analysis of information.
(+) Temporary storage to obtain better results in the analysis of the information.
(+) Extraction of patterns using artificial intelligence algorithms.
(-) A data science specialist may not have knowledge in the application domain.
(-) Learning in data science takes time and can increase the cost of the project.
(-) Learning to know how to apply artificial intelligence algorithms and use the analytics tools also takes time.

6.7 Known Uses

[Souza et al. 2013] present patterns that help both novice and experienced data scientists to discard invalid bug data that could lead to wrong conclusions. Sipina[3] and Weka[4] are examples of tools that can help team members to develop some skills in data science. [Kim et al. 2016] present a training agenda for data scientists and describe their missions in software engineering contexts, identifying five distinct working styles of data scientists.

7. CHOOSE THE RIGHT PARTICIPANTS

7.1 Motivational Example

Continuous experimentation (CE) in business is primordial for growth and continuous improvement aspects, as demonstrated in the cases of Amazon and Microsoft. For this purpose, the selection of people who dominate

[3] http://eric.univ-lyon2.fr/ ricco/sipina.html
[4] https://www.cs.waikato.ac.nz/ml/weka/

specific subjects in each stage of the project is necessary so that the obtained results are in agreement with the target public. The wrong choice of people to participate in the experimentation project may cause an unnecessary expenditure of resources and increased risk in decision making.

7.2 Context

There is a need for qualified people at each stage of the experiment. To perform experiments, it takes people with different abilities to assist in a positive way. Research within the process of CE in organizations depends not only on structured tools and programs but also on people capable of performing tasks and analyzing their processes,in both quantitatively and qualitatively. In the stages of CE is common to segment the processes to achieve the proposed objective effectively. For this, it is necessary to choose the right people for the whole process of experimentation.

7.3 Problem

How to identify and select the right people to participate in the experiment?

7.4 Forces

- Professionals who define the experiments and choose the people participating in them must also have a high knowledge of the project in all stages and methodologies of experimentation for assertive choice.
- Not all companies have internal teams with competence to participate in the experimentation process.
- Choosing the right people for the experiment is key to aggregating and achieving the objectives of the experiment.
- People who do not have expertise in the experimental area may not contribute to the success of experimentation.
- Comparisons with new teams and people outside the organization can assist in the process of measuring results.

7.5 Solution

Choose groups of people internal and external to the organization, based on the specific domain, leveling of knowledge and considering skills that can contribute to each phase of the experiment.

Choosing the right people implies knowing the characteristics of the target audience and the goals of the project. Thus, the profile of the participants can vary according to the project, or even according to the phase of the project. The participants will be selected according to their abilities and skills, and the purpose of the experiment.

7.6 Consequences

(+) The experiment passes by competent people to perform specific tasks of their knowledge, making the results more efficient and objective.

(+) It will be possible to analyze qualitative data, streamlining the decision-making process.

(+) The interpretation of the experiment as well as the research data becomes more appropriate and focused on the experiment.

(+) It may contribute to greater internal competition among professionals for the participation of experiments according to their individual abilities.

(-) People who are not selected to participate in the experiment team may feel harmed or belittled.

7.7 Known Uses

Sometimes, the customer feedback loop is slow and there is often a lack of mechanisms that allow an efficient data collection and analysis. In this case, the challenges of product management is to obtain accurate and timely feedback. To resolve this issue, [Olsson and Bosch 2014b] develop a model called HYPEX (Hypothesis Experiment

Data-Driven Development) that supports companies in running feature experiments to shorten customer feedback loops. [Olsson and Bosch 2014a] argue that teams could respond faster and act more pro-actively towards customers.[Fagerholm et al. 2017] presented The RIGHT model for CE that describes the experimentation process where the infrastructure architecture for supporting the model takes into account the roles, tasks, technical infrastructure, and information artifacts needed to run large-scale continuous experiments.

8. MOST SUITABLE WAY

8.1 Motivational Example

After planning for some change or development in a particular software application, you need to check and evaluate how best to execute this planning and make the best decisions. The decisions made through the results of the experiments allows the analysis in a criteria list related to how the decision should be made in the most suitable way. This CE pattern can be applied to the developer and customer in order to generate feedback based on the data from experiments.

8.2 Context

Develop experiments to verify how to apply in the best way certain premise (decision/hypothesis). Plan an intervention or treatment for a certain software feature and check in a criteria list that validates this intervention what is the best to follow. The created experiments aim to generate data to verify which criteria have the best acceptance. In this sense, in order to do software changes for performance improvement or customer acceptance, it is necessary to define a series of possible ways to carry out the proposed changes.

8.3 Problem

How to find what is the most suitable way or the best criterion to confirm the proposed change?

8.4 Forces

—Without experiments, the decision on what is the best for the application can be taken in the wrong way.

—Decisions will be based on data, not on people’s opinions.

—It is possible to optimize the decision-making process by looking at the responses of each item on the list. In this way, you can see how best to change the software or application.

8.5 Solution

After creating the hypotheses, plan a series of ways that can confirm these hypotheses. First, a list of criteria/ ways is developed indicating which methods can execute the initial idea. After the definition of the possible ways is carried out the experimentation.

Experimentation will generate data that, when interpreted by the data scientist, will indicate the suitability of the ways. Must be on the list, a default way, which allows you to check if all previous ways that have low acceptance. If the default way has the highest percentage of acceptance then the initial ways should be remodeled.

Example of use: The hypothesis is to modify the purchasing system of an e-commerce site to better serve customers. First, define a list of ways to solve this problem, For example:

—Way 1: Modify the sales panel;

—Way 2: Modify the “Buy” button;

—Way 3: Modify sales options (card, ticket, Internet Bank);

—Way 4: Default, None of the Previous Ways.

After creating the list of possible ways, develop the experiments and generate data for each them to find which way has achieved the best performance. To help you to figure out which is the most suitable way, you may need help the assistance from the data scientist. If the default way gets the highest score, then the other ways should be remodeled.

Figure 2 shows the construction of the criteria/ways list, indicating the possible ways to test the defined hypothesis. In the end, is chosen the way with the best percentage, or acceptance rate.

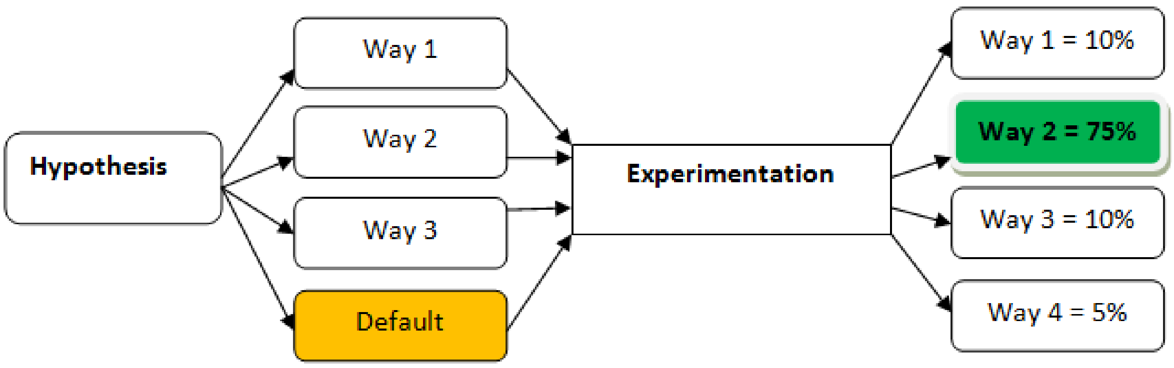

Fig. 2. Scheme of possible ways to test a hypothesis.

8.6 Consequences

(+) The developer can check for minor changes to the software.

(+) The results of the experiments make it possible to observe the behavior of the users who use the application.

(+) Deciding on a list of criteria reduces the possibility of performing software modification in the wrong way.

(-) Creating an list of ways can be very complex.

(-) Misinterpretation of experiments can lead to wrong decisions.

(-) There may be abrupt changes in development when none of the experiments can observe suitability to the list.

8.7 Known Uses

Example of the Amazon website [Kohavi et al. 2014], which starts a series of experiments to verify how best to serve customers, i.e, is observed a search for solutions to execute the decision in the best way. The Optimizely tool [Siroker et al. 2014] aims to generate experiments and tests to verify patterns of customer behavior to improve applications on websites. In this same segment, [Liu and Chamberlain 2018] proposed the use of online controlled experiments to verify the impact of the changes in software and web applications.

9. EXPERIMENTATION KNOWLEDGE BASE

9.1 Motivational Example

Consider a company that uses CE and performs numerous experiments on the product/service during their development life cycle. These experiments need to be documented to have a knowledge base of all experiments, types of experiments, target audience and the results. Otherwise, similar experiments can be implemented in other applications with reduced resources and time.

9.2 Context

On the CE, there's a diversity of tests types, as for example, A/B test [5], canary test[6], and others which are often performed. Not all experiments are successful, but the importance of their results and how they were made is equally important to the life cycle of a product/service and needs to be documented.

This systematization of the experiments allows improvements in a product/service to be made in a systematic way and not through unsupported assumptions and opinions. This is done from data and feedback from the client/user, avoiding unnecessary risks in the decision making, thus creating a knowledge base, i.e., a history of experimentation to assist future experiments.

9.3 Problem

How to document the tests made in CE, in order to make them easy to understand, retrieve and also detailed?

9.4 Forces

- Not every experiment has positive results or is well executed, but its documentation is important.
- If an experiment fails, it is good to know of its occurrence, how it occurred and the user feedback.
- Not every experiment has direct user feedback, part is obtained through usage data.
- The register of the experiments must not only have the results, but also the modifications and/or addition of features and the metrics as an expected result.
- In addition to the results of the experiment, the costs for its realization and the material and work resources used must be recorded.
- Decisions based on the experiments should be recorded and whether future results were satisfactory.

9.5 Solution

Document the experiments and their results in order to create a knowledge base for CE process that allows future consults and the replication of new experiments.

By documenting experiments and their results, the team can analyze because it was not possible to validate the results (failed experiments); or because it did not provided expected results (misused experiments). And, with regard to experiments performed successfully, at any time, the team can check the actions that have been taken. The propose this pattern is to store all instances of the continuous experimentation process in order to allow future data analysis, or to be a basis for future experiments. Table I presents some examples of the experiments information that can be stored in a knowledge base.

9.6 Consequences

(+) It is possible to analyze the history of experiments performed on a system and to know the decisions taken during the lifetime of the system.

(+) Check for successful experiments and make new ones based on them.

(+) Avoid repeating failed experiments.

(-) More time is needed to document the experiments.

(-) Discourage repeating experiments that have not been successful in the past, but which could be interesting at the present time.

[5] A/B test is a way of testing that compares two versions (version A and version B), to decide what is better [Bakshy et al. 2014].

[6] Canary test is a technique to reduce the risk of introducing a new software version in production by slowly rolling out the change to a small subset of users before rolling it out to the entire infrastructure and making it available to everybody [Shahin et al. 2017].

Table I. Useful information about experiments. Adapted from [Munezero et al. 2017]

Data	Description	Example
Experiment Name	Brief title that describes the experiment.	Choice of related products in the shopping cart.
Assumption	Represents an idea that is accepted as truth without proof.	Recommending different products in relation to the shopping cart generates more purchases than similar products.
Hypothesis	The hypothesis explains a phenomenon.	We believe that offering related products from a different category will have a 20% increase in sales with more than one item.
Experiment Plan	Is the planning of how to test a hypothesis, including the type of the test, how the test was done, experiment object (represents critical aspects of the product that will be tested), collect (results of the collected experiments as qualitative data as use data or quantitative as interviews, observations and surveys) and data analysis (verification for validation of a hypothesis).	An A/B test will be carried out in two weeks in which products of a different category will be offered for customers with more than two years of registration on the site and will measure the sales in that period.
Experiment Object	Represents critical aspects of the product that will be tested.	Section of related products on the sales site.
Testers	Are the audience that will participate in the experiments.	Customers with more than two years of registration on the site.
Metrics	Represent the capture of the values of the product or resource at a specific time of data collection.	Sales with products offered in the same category compared to sales with products of different categories carried out in the period of two weeks.
Success criteria	Analyze the metric that allows verifying if the hypothesis is correct.	Sales increase of 20%.
Decision Taken	The decision to be made based on the data. Can apply the idea, to rethink because it needs to be better worked or cancel because there is no point to continue.	Based on the 23% increase, the idea seems to be good and should be applied to a larger audience in a new experiment.

9.7 Known Uses

Tools such as VALUE ([Munezero et al. 2017]), use visualization mechanisms that aid in decision making, as shown in the Figure 3. Currently, there are several tools that aid Knowledge Discovery in Databases (KDD) from the results of each experiment in order to create relationships of interest. KDD is a general concept of a knowledge extraction process from databases for identifying comprehensible, valid, new and potentially useful patterns from large data sets [Fayyad et al. 1996]. Table II presents some operational tools for KDD, according to [Passos and Goldschmidt 2005].

10. CONCLUSION

In this paper, we have presented some patterns related to the continuous experimentation process. These patterns were identified in the existing literature about CE, from examples of companies that have successfully carried out continuous experimentation to improve its products. Next, we present a summary of the patterns:

—**Pilot Experiments:** Choose small teams, create simple experiments that aim to test small parts of the product.

—**Data Scientists Skills:** Hire a professional or train a team member to play the data scientist role within the continuous experimentation team.

—**Choose the Right Participants:** Choose groups of people internal and external to the organization to meet the target audience interests.

—**Most Suitable Way:** After creating the hypotheses, plan a series of ways that can confirm these hypotheses.

—**Experimentation Knowledge Base:** Document the experiments and their results in order to create a knowledge base for continuous experimentation process.

Table II. Operational tools for KDD. Adapted from: [Passos and Goldschmidt 2005]

Name	KDD Tasks	Source
SPSS/Clementine	Classification, Association Rules, Clustering, Sequential Pattern Analysis, and Desviation Detection.	SPSS Inc. (www.spss.com)
Poly/Analyst	Classification, Regression, Association Rules, Clustering, Summarization, and Desviation Detection.	Megaputer Intelligence (www.megaputer.com)
Weka	Classification, Regression, Association Rules, and Clustering.	University of Waikato (www.cs.waikato.ac.nz/ml/weka)
Intelligent Miner	Classification, Association Rules, Clustering, Summarization, and Sequential Pattern Analysis.	IBM Corp. (www.ibm.com)
WizRule	Classification, Summarization, and Desviation Detection	WizSoft Inc. (www.wizsoft.com)
Bramining	Classification, Regression, Association Rules, and Summarization.	Graal Corp. (www.graal-corp.com.br)
SAS Enterprise Miner	Classification, Regression, Association Rules, and Summarization.	SAS Inc. (www.sas.com)
Oracle Data Mining	Classification, Regression, Association Rules, Clustering, and Text Mining.	Oracle (www.oracle.com)

Fig. 3. Value tool [Munezero et al. 2017]

11. ACKNOWLEDGMENTS

We would like to thank our shepherd Hernán Astudillo for his valuable comments and feedback. Furthermore, we would like to thank our 2018 SugarLoaf PLoP Writers Workshop Group for their valuable comments and suggestions. Finally, we would like to thank the support granted by Brazilian funding agency CAPES.

REFERENCES

Eytan Bakshy, Dean Eckles, and Michael S. Bernstein. 2014. Designing and Deploying Online Field Experiments. *CoRR* abs/1409.3174 (2014). `http://arxiv.org/abs/1409.3174`

Jan Bosch and Ulrik Eklund. 2012. Eternal Embedded Software: Towards Innovation Experiment Systems. In *Leveraging Applications of Formal Methods, Verification and Validation. Technologies for Mastering Change*, Tiziana Margaria and Bernhard Steffen (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 19–31.

Aleksander Fabijan, Pavel Dmitriev, Helena Holmström Olsson, and Jan Bosch. 2017. The Evolution of Continuous Experimentation in Software Product Development: From Data to a Data-driven Organization at Scale. In *Proceedings of the 39th International Conference on Software Engineering (ICSE '17)*. IEEE Press, Piscataway, NJ, USA, 770–780. `DOI:http://dx.doi.org/10.1109/ICSE.2017.76`

Fabian Fagerholm, Alejandro Sanchez Guinea, Hanna Mäenpää, and Jürgen Münch. 2014. Building blocks for continuous experimentation. In *Proceedings of the 1st international workshop on rapid continuous software engineering*. ACM, 26–35.

Fabian Fagerholm, Alejandro Sanchez Guinea, Hanna Mäenpää, and Jürgen Münch. 2017. The RIGHT model for Continuous Experimentation. *Journal of Systems and Software* 123 (2017), 292 – 305. `DOI:http://dx.doi.org/https://doi.org/10.1016/j.jss.2016.03.034`

Federico Giaimo and Christian Berger. 2017. Design Criteria to Architect Continuous Experimentation for Self-Driving Vehicles. *2017 IEEE International Conference on Software Architecture (ICSA)* (2017), 203–210.

David Issa Mattos, Jan Bosch, and Helena Olsson. 2018. Challenges and Strategies for Undertaking Continuous Experimentation to Embedded Systems: Industry and Research Perspectives. (05 2018).

Natalia Juristo and Ana M. Moreno. 2010. *Basics of Software Engineering Experimentation* (1st ed.). Springer Publishing Company, Incorporated.

Katja Kevic, Brendan Murphy, Laurie Williams, and Jennifer Beckmann. 2017. Characterizing experimentation in continuous deployment: a case study on bing. In *Proceedings of the 39th International Conference on Software Engineering: Software Engineering in Practice Track*. IEEE Press, 123–132.

Miryung Kim, Thomas Zimmermann, Robert DeLine, and Andrew Begel. 2016. The emerging role of data scientists on software development teams. In *Proceedings of the 38th International Conference on Software Engineering*. ACM, 96–107.

Ron Kohavi, Alex Deng, Brian Frasca, Toby Walker, Ya Xu, and Nils Pohlmann. 2013. Online Controlled Experiments at Large Scale. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '13)*. ACM, New York, NY, USA, 1168–1176. `DOI:http://dx.doi.org/10.1145/2487575.2488217`

Ron Kohavi, Alex Deng, Roger Longbotham, and Ya Xu. 2014. 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, New York, NY, USA, 1857–1866. `DOI:http://dx.doi.org/10.1145/2623330.2623341`

Ron Kohavi, Roger Longbotham, Dan Sommerfield, and Randal M. Henne. 2009. Controlled Experiments on the Web: Survey and Practical Guide. *Data Min. Knowl. Discov.* 18, 1 (Feb. 2009), 140–181. `DOI:http://dx.doi.org/10.1007/s10618-008-0114-1`

Thomas Kuhn. 1981. *What are scientific revolutions*. Boston: MIT.

Eveliina Lindgren and Jürgen Münch. 2016. Raising the odds of success: the current state of experimentation in product development. *Information and Software Technology* 77 (2016), 80–91.

CH Liu and Benjamin Paul Chamberlain. 2018. Online Controlled Experiments for Personalised e-Commerce Strategies: Design, Challenges, and Pitfalls. *arXiv preprint arXiv:1803.06258* (2018).

Daniel Lucrédio, Eduardo Santana De Almeida, Alexandre Alvaro, Vinicius Cardoso, and Eduardo Kessler Piveta Garcia. 2004. Student's PLoP Guide: A Pattern Family to Guide Computer Science Students during PLoP Conferences. In *Proceedings of the Latin American Conference on Pattern Languages of Programs (SugarLoafPLoP)*.

Myriam Munezero, Sezin Yaman, Fabian Fagerholm, Petri Kettunen, Hanna Mäenpää, Simo Mäkinen, Juha Tiihonen, Leah Riungu-Kalliosaari, Antti-Pekka Tuovinen, Markku Oivo, and others. 2017. Continuous Experimentation Cookbook. (2017).

Helena Holmström Olsson and Jan Bosch. 2014a. Climbing the "Stairway to Heaven": evolving from agile development to continuous deployment of software. In *Continuous software engineering*. Springer, 15–27.

Helena Holmström Olsson and Jan Bosch. 2014b. The HYPEX model: from opinions to data-driven software development. In *Continuous software engineering*. Springer, 155–164.

Emmanuel Passos and Ronaldo Goldschmidt. 2005. Data Mining: a practical guide. *Editora Campus, Rio de Janeiro (in Portuguese)* (2005).

O. Rissanen and J. Münch. 2015. Continuous Experimentation in the B2B Domain: A Case Study. In *2015 IEEE/ACM 2nd International Workshop on Rapid Continuous Software Engineering*. 12–18. DOI:http://dx.doi.org/10.1109/RCoSE.2015.10

Gerald Schermann. 2017. Continuous experimentation for software developers. In *Proceedings of the 18th Doctoral Symposium of the 18th International Middleware Conference*. ACM, 5–8.

Mojtaba Shahin, Muhammad Ali Babar, and Liming Zhu. 2017. Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices. *CoRR* abs/1703.07019 (2017). http://arxiv.org/abs/1703.07019

Dan Siroker, Pete Koomen, Elliot Kim, and Eric Siroker. 2014. Systems and methods for website optimization. (Sept. 16 2014). US Patent 8,839,093.

Rodrigo Souza, Christina Chavez, and Roberto Bittencourt. 2013. Patterns for cleaning up bug data. In *Data Analysis Patterns in Software Engineering (DAPSE), 2013 1st International Workshop on*. IEEE, 26–28.

Diane Tang, Ashish Agarwal, Deirdre O'Brien, and Mike Meyer. 2010. 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, New York, NY, USA, 17–26. DOI:http://dx.doi.org/10.1145/1835804.1835810

Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in software engineering*. Springer Science & Business Media.

Sezin Gizem Yaman, Myriam Munezero, Jürgen Münch, Fabian Fagerholm, Ossi Syd, Mika Aaltola, Christina Palmu, and Tomi Männistö. 2017. Introducing continuous experimentation in large software-intensive product and service organisations. *Journal of Systems and Software* 133 (2017), 195–211.