MASTER'S THESIS 2021

A/B testing Customer Admin — an empirical validation of controlled experimentation on internal tools

Amalia Paulsson

ISSN 1650-2884
LU-CS-EX: 2021-18

DEPARTMENT OF COMPUTER SCIENCE
LTH | LUND UNIVERSITY

EXAMENSARBETE
Datavetenskap

LU-CS-EX: 2021-18

A/B testing Customer Admin — an empirical validation of controlled experimentation on internal tools

A/B-testning av Customer Admin — en empirisk validering av kontrollerade experiment på interna verktyg

Amalia Paulsson

A/B testing Customer Admin — an empirical validation of controlled experimentation on internal tools

Amalia Paulsson
am3817pa-s@student.lu.se

June 4, 2021

Master's thesis work carried out at
the Department of Computer Science, Lund University.

Supervisors: Rasmus Ros, rasmus.ros@cs.lth.se
Joakim Månsson, joakim.mansson@ingka.com

Examiner: Per Runeson, per.runeson@cs.lth.se

Abstract

The usage of A/B testing and other controlled experimentation methods in the online setting is growing globally as companies are more prone to make data-driven decisions from real-world user feedback. Previous research has contributed with domain specific validations of controlled experimentation, identifying challenges and benefits along with other aspects that play a critical role for the success of its implementation.

In this study, the primary goal is to validate A/B testing in the internal domain, i.e. online services exclusively used by company employees.

This empirical validation is presented through a proof of concept implementation on Customer Admin, a tool that helps approximately 34 500 IKEA coworkers to interact with customer data. The implementation comprised stakeholder interviews to understand objectives and ensure that a pertinent hypothesis was phrased, test execution where data was collected and processed for 33 days, version evaluation and complementary user questionnaires.

The results suggest that simplicity and time efficiency are key objectives for the user. Moreover, data collected on the key metric was too scarce to allow rejection of the null hypothesis of version A and B performing equal. Nonetheless, secondary metrics and additional user questionnaire suggest that the users are more efficient in the new menu design and that the users prefer it to the old.

Conclusively, the net utility of A/B testing internal tools is explored by comparing the value of quantitative user feedback to the cost of implementation. One main cost driver is the customization of tool-specific objectives and metrics. More research is needed in order to quantify these terms and further expand the domain-specific knowledge in A/B testing.

Keywords: A/B testing, Online Controlled Experimentation, Hypothesis Engineering, Internal Tools, User Behavior

Acknowledgements

This thesis has been conducted at the Department of Computer Science at Lund University, in collaboration with INGKA Holding B.V. The thesis is the last course before completing a Master's degree in Industrial Engineering and Management.

I would like to thank my supervisor and Senior Software Engineer at INGKA Joakim Månsson, who has helped and worked along with me throughout this project. Without his guidance and effort, this mission would not have been accomplished. I would also like to give a big thank you to Engineering Manager Magnus Pettersson, and all the team members in Customer Admin. Lastly, I would like to extend my gratitude to Rasmus Ros, my supervisor at LTH, for valuable input and continous feedback.

Contents

Chapter 1
Introduction

The interest for finding trustworthy conclusions through data is growing globally and A/B testing is one of the Internet industry's most commonly used methodologies for large-scale experimentation [13]. An A/B test is defined as a simple setup of a controlled experiment — an experiment used to find probable causal relationships by randomly splitting subjects between versions and instrumenting behaviors to determine some evaluative metric [15]. The value of deploying this in the online setting lays in the unveiled insights into the actual user behaviour, which helps software developers to quickly evaluate design ideas and nonetheless expand their knowledge about their users.

Nevertheless, even though large actors in the internet industry such as Microsoft, Facebook and Google started applying A/B testing a decade ago, it is still considered to be in an early stage of development [26]. As of today, research in online controlled experimentation focuses on external websites and tools, that is commercial services where the users are customers to the company in question. Meanwhile there are multiple types of services that are exclusively utilized within organizations, aiming to assist co-workers in managing their tasks. These internal tools evolve continuously and can be critical to the company's operational success.

IKEA have developed such a tool called Customer Admin, that allow IKEA co-workers to interact with customer data. The development team of Customer Admin, however, find it challenging to establish a user-centered approach to the design- and development process and a curiosity has evoked for how A/B testing can be incorporated into their development process. But since online controlled experimentation is a scientific method and should be based on empirical research, the lack of academic support for A/B testing's applicability on internal tools poses a problem.

Therefore the purpose of this project is to evaluate A/B testing as a solution instance on the development of internal tools through an implementation on Customer Admin.

RQ: How can A/B testing be applied in the development of internal tools?

To answer the research question, a proof of concept A/B test was implemented in Customer Admin, providing a practical example and a base for discussion. The method can be

divided into three overaching steps: Firstly interviews were held with various stakeholders which laid the groundwork for forming a hypothesis that addressed the actual goal of a new design element. Thereafter the experiment was launched, testing a new start menu by comparing it with the original and user interactions were tracked for 33 days using Google Analytics. The aggregated data was analysed through post-collection processing script, written in Python, that calculated the evaluative metrics and associated statistics. As a complement to the A/B test, questionnaires about the user experience were sent out and answered by 9 users. The answers where used to compare the test results with the users own preferences.

The interviews revealed user efficiency as the main objective of the tool, i.e. its ability to assist the co-workers in finishing their respective tasks as fast as possible. Moreover clarity and simplicity was stressed, making it intuitive for the co-workers to find what they are looking for. From the insights created during the interviews, a hypothesis was phrased for the proof of concept implementation:

"We predict that a new start menu for co-workers in the Australian market will shorten the average time spent per customer errand because it makes the orientation in Customer Admin easier to comprehend. We will know this is true when we see a decreasing time difference from entering a customer profile and saving a change".

The calculated statistics, however, proved the data to be too scarce to either let the null hypothesis be accepted nor rejected. Yet, secondary metrics as well as the complementary questionnaires promoted the new menu design, making the users more efficient and advocating their own preferences.

Conclusions drawn emphasised the scarce data due to relatively low traffic as a general challenge when A/B testing internal tools. In order to mitigate the consequences of low traffic, larger test groups and more frequently triggered metrics for evaluation was proposed. Moreover sources of error in the used metrics were discussed and the conflict between having independent test users for the sake of statistics and maintaining a coherent working culture were co-workers communicate and collaborate. More research is needed before sufficiently declaring A/B testing's applicability in the internal domain. Focus should be kept on enhancing rigour by gathering more practical case studies on other types of internal tools.

As described in the preceding list of contents, the report starts by introducing some background knowledge within A/B testing as well as presenting Customer Admin, as the tool to be tested. Thereafter chapter 3 lays out the method used, from the format of the interviews to the test setup in customer admin, post collection analytics and the complementary questionnaires. Then chapter 4 presents and briefly evaluates the results from the interviews, the A/B test and the questionnaires. Finally in chapter 5 key take aways concerning the Customer Admin tool as well as A/B testing internal tools in general are discussed for the purpose of answering the research question.

Chapter 2
Background

This chapter presents relevant background knowledge about the A/B testing methodology and Customer Admin on which the A/B testing pilot will be deployed. Among other available sources of frameworks on how to implement an A/B test, this chapter will focus on the outline presented by King et al. [13]. Since they provide a simple guide on how to get started with A/B testing, it was considered a suitable source of information when introducing A/B testing in a new context.

2.1 A/B testing

In an A/B test the controlled experimentation is manifested online by taking two versions of a software, the control (A) version which is usually the default version and the treatment (B) which contains a change [15], see Figure 2.1. This method is often applied to evaluate software updates that are believed to have a positive impact on user behavior, assigning one user group to the already existing version and one user group to the new version — i.e. the control and the treatment respectively. The primary advantage of using A/B testing in decision making is that it brings insights into the real-world user experience through a direct feedback loop with the users [13]. Giving developers access to continuous flows of data from this feedback loop increases their ability to detect and fix problems, as well as simplifying code and removing unnecessary features [2].

The experimental approach that builds on hypotheses is, however, not the only way to adapt user experience in software development. Melegati et al. [19] compares experiment-driven development with requirement-driven, see Figure 2.2 which they claim is the traditional approach to software development. In requirements engineering (RE) the development have been fueled by requirements specified by product management or clients. Although RE unveils the software's intended purpose and identifies stakeholders and their demands, the features needed by the users are built from user stories which provide limited insight into the de facto user demands. In experiment engineering (EE) on the contrary, the primary goal is

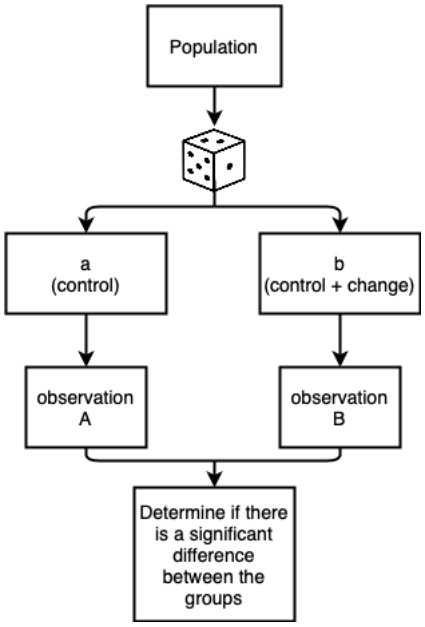

Figure 2.1: In an A/B test, a change is observed by allocating a population on a control group and treatment group. Differences between the groups are then measured, showing the effect of the change.

to learn about the user rather than the code itself. Therefore Melegati et al. [19] promote EE in contrast to RE and present guidelines on how to identify, prioritize and verify hypotheses, making the development process somewhat amenable to information that is collected as the software evolves. King et al. [13] imply that the A/B testing process can be divided into three phases:

- **The definition phase** which is about creating a hypothesis that efficiently encapsulates a relevant problem.
- **The execution phase** which is about setting up the experiment and put it in action.
- **The analysis phase** which is about interpreting the results and make conclusions on how the different versions perform.

Based on these three phases King et al. [13] (pp 90) introduce a framework for the experimentation of A/B testing, see Figure 2.3, where the phases are broken down into steps forming a tree structure. The definition phase involves goals, problems and hypothesis generation, the execution phase involves hypothesis design and test, and the analysis phase involves test and result. Throughout the first four levels there are flows of data that continuously feed in new information resulting in knowledge about what problems/opportunities are worth exploring and how the experiments themselves are fulfilling their purpose.

2.1.1 The Definition Phase

This phase is about planning the experiment so that identifying problem and opportunity areas and creating hypothesis address the overall goal. Therefore the first step in the definition

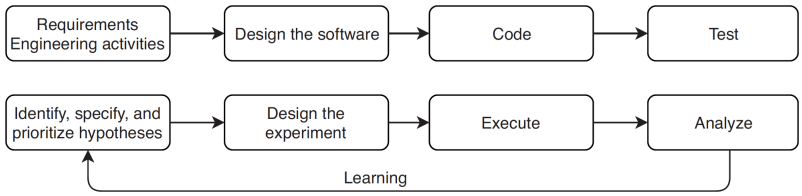

Figure 2.2: Comparison of requirement-driven and experiment-driven software development by Melegati et al. [19].

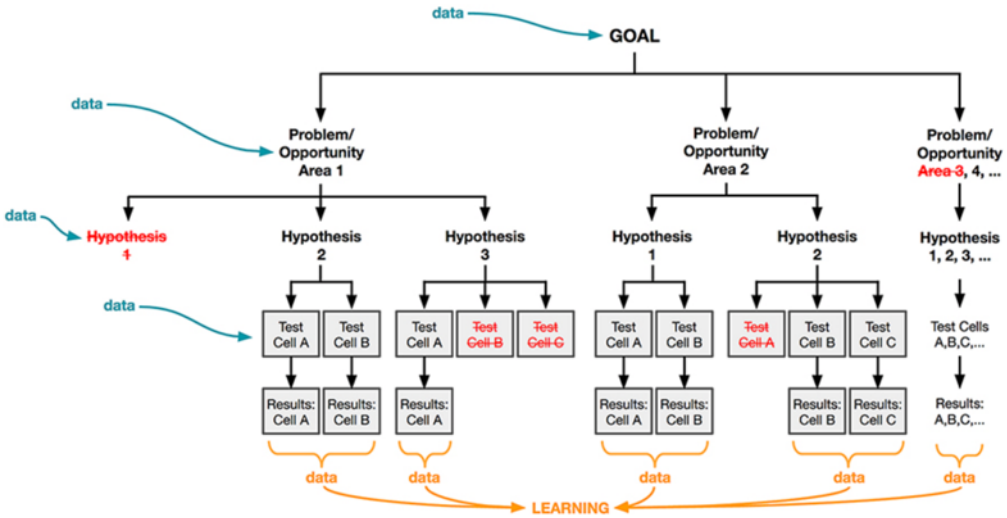

Figure 2.3: The A/B testing methodology tree [13].

phase is to define the goal. “Without goals, and plans to reach them, you are like a ship that has set sail with no destination.” – Fitzhugh Dodson. In order to find answers with a qualitative depth from the experiment, King, Churchill and Tan [13] emphasize the importance of maintaining collaborative relationships with product owners, managers, users, and other stakeholders during this step. All of their viewpoints should be picked up and included in the goal definition. On the contrary, Melegati, Wang and Abrahamsson [19] recommend excluding the user’s opinions in the definition phase. Even though it is important to understand the users’ desires, in hypothesis engineering the point is to learn about the users through the hypothesis rather than through questionnaires and interviews. This is one of the reasons why hypothesis engineering is considered more time efficient. Hereof Melegati, Wang and Abrahamsson stress that evaluation of the hypothesis should be avoided before the launch of the A/B test and to only to consult with product management and developers in this phase. Seeing the essence in both King et al. and Melegati et al.’s reasonings, Johanssen, Reimer and Bruegge [11] propose continuous thinking aloud (CTA), an approach where verbal user reviews are collected as the application is in production. This gives the developers access to descriptive feedback and in more regular loops. The performance of CTA is, however, highly dependent on the accuracy and performance of the speech and word processing that categorized the feedback. According to King, Churchill and Tan a good goal should address the following questions [13].

- What is believed to be best for the users?
- Where are time and resources wished to be allocated?
- What are the business and user-critical problems that could form opportunities or problems?
- What are the biggest opportunities in terms of improving the user experience?

These questions, together with available data shown in Figure 2.3, should facilitate finding a goal that supports a desired impact on the user experience. Since there are multiple determining factors for a user experience, King et al. [13] imply that a goal can either be qualitative, quantitative or both. However, in order to determine whether a goal is achieved or approached it should be measurable. This also enables choosing thoughtful metrics that efficiently tracks the progress towards these goals. A metric must be measurable and should clearly articulate a user behavior that is to be influenced. Notwithstanding the metric's sensitivity should not be too low, that is how much a change in experience causes the metric to change.

King, Churchill and Tan presents three types of metrics: the metric of interest, that determines failure or success of the test, key metrics, that help detecting negative side effects, and secondary metrics, that creates a contextual understanding and can help detecting positive side effects of the test [13]. Munaiah and Meneely introduce an another type of metric they call vulnerability metrics that help developers discover vulnerabilities and mistakes directly in the code. Out of a literature study [20], they compile ten metrics: the number of lines modified in a file, the number of developers collaborating on the same files, the number of commits, the cyclomatic complexity, number of previous vulnerability fixes, number of recurrent functions, file sizes, number of input parameters in functions, number of outputs and the number of unique decision paths.

When goal and metrics are defined, the next step is to find approaches to achieve the goal, illustrated as problem/opportunity areas in Figure 2.3. These should be areas that can be innovated toward the goal and should link back to the metrics of interest. Nonetheless, since one problem/opportunity area should generate several hypotheses, it is favorable to go broad at this stage. The problem/opportunity statement should address the biggest problem with your user experience that are impeding the goals or the biggest opportunity for a desired effect or improvement. Hence data will again play a crucial role in identifying authentic areas. The effect reflected in the problem/opportunity area will become a critical component in the hypothesis. A hypothesis is a prediction of what will happen to the users when introducing the change. King, Churchill and Tan propose a framework for generating a strong hypothesis: *We predict that [change] for [user group(s)] will achieve [effect] because of [rationale]. We will know this is true when we see [measure]* [13]. Where user group is the population or subset of users, change is what will be added to the control experience to impact user behavior, effect is the desired impact of the change, rationale is a motivation for why the hypothesis is a sensible prediction, and measure is the metrics hoped to impact. Even though these five elements are claimed to be important in a hypothesis, using all of them is not necessary in every occasion. At least, change, effect and measure should be included [13].

2.1.2 The Execution Phase

This phase is about building test cells out of the hypothesis that encapsulate the experiment. As each cell stages a unique way of approaching the hypothesis, the first step is to design the hypothesis. This means developing a version that is representative of the hypothesis and that will give results with useful information. As shown in Figure 2.3, oftentimes one hypothesis result in multiple test cells. However, this is not always the case. In some experiments where a hypothesis only considers one small element, it is more logical to only build one test cell out of that hypothesis.

King et al. argue there are two dimensions in a design process: scope and level of fulfillment [13]. The scope can be global or local, where global means having several variables in a test cell and local means having only one or two. Therefore, global experiments often include significantly different designs whereas local experiments are less varying. The level of fulfillment refers to the recent step in the process of addressing the problem/opportunity area. It can either be in a explorative or evaluative state, where exploration is an early stage where the experiments are being crafted and evaluation is a later stage where small adjustments are made to establish strong causality [13].

Data can be collected longitudinally or in a snapshot. Longitudinal data comes from one or a few users over a period of time and shows how users learn and adapt to changes, whereas snapshot data is collected through observations of multiple users in one instant [13].

2.1.3 The Analysis Phase

The analysis phase is about launching the A/B test, collecting data and getting out the information that is needed to draw trustworthy conclusions. The mechanics and close procedure of the launch is particular to the product. Before data is collected, King, Churchill and Tan state that the minimum detectable effect (MDE) should be defined. This is the lower limit of the increase or decrease in the metric of interest where the test is considered successful. Accordingly the tests need to be designed with the power to display differences at least as big as their respective MDE. Power is in this context decided by sample size, confidence level and variance [13]. As for vulnerability metrics, Munaiah and Meneely argue the definition of thresholds after data has been collected. Since these metrics measure contextual side effects, they are presumably project specific, and it is therefore difficult to set thresholds in advance. Instead they define thresholds as quantiles, dividing the observed metric values into different levels of risk: 70%–80% is regarded as a medium risk, 80%–90% a high risk and 90% and up a critical risk [20].

As the A/B test is rolled out, awareness should be kept about some tradeoffs such as between the number of test users and testing time. The more users included in the A/B test, the faster it will take to collect a sufficient result. Yet with more users included, more user experiences is at stake during the launch. Another tradeoff is the sample size versus the significance level. The larger the sample size is the larger the significance level can become. But then again the larger test group jeopardizes more user experiences, especially if the test is global [13].

Kohavi et al. [14] claim there are two formulas that are relevant when evaluating whether the treatment is different than the control. Firstly a t-test to evaluate the null hypothesis of that the mean of the metric of interest in the treatment and the control are the same. In

this project the number of samples and consequently also the variance was different in the control and the treatment. Thus Welch's t-test was applied, see equation 2.1, where μ_B and μ_A are the mean values of the control and the treatment's respective metrics of interest, σ_A and σ_B are the respective standard deviations, and n_A and n_B are the sample sizes [27]. Based on the test result $|t|$, the hypothesis is compared with a threshold value of t, e.g. 1.96 for 95% confidence. If $|t|$ is larger than the threshold the null hypothesis is rejected and claim that there is a difference between control and treatment.

$$t = \frac{\mu_B - \mu_A}{\sqrt{\frac{\sigma_A^2}{n_A} + \frac{\sigma_B^2}{n_B}}} \tag{2.1}$$

The second formula Kohavi et al. [14] stipulates useful in the context of hypothesis evaluation is the calculation of the minimum sample size, see equation 2.2. n is the number of observations in each variant, σ is the standard deviation of the of the metric, and Δ is the absolute change to be detected. Moreover a 95% confidence level is assumed and a statistical power of 80%, that is the probability of correctly rejecting the null hypothesis.

$$n = \frac{16\sigma^2}{\Delta^2} \tag{2.2}$$

2.1.4 Continuous Experimentation

The experiment is effectively recognized as a sequence of events where the result from one test might be decisive to the execution of a following test. However, it does not have to be a linear process and steps that are somewhat independent are usually performed in parallel [13]. Fagerholm et al. [6] defines the building blocks for continuous experimentation, see Figure 2.4, where the technical infrastructure is developed in parallel with reoccurring Build-Measure-Learn blocks. The blocks illustrate activities performed when conducting a controlled experiments and could be related to King et al.'s [13] three phases: design, execution and analysis. In accordance with Melegati et al. [19], Fagerholm et al. [6] argue that despite the risk of degrading user experience, negative experiments should also be run since the primary objective of learning about the user has long-term benefits.

In a later work [7] Fagerholm et al. present a systematic framework model, named the RIGHT model, for continuous experimentation that display the roles, tasks, technical infrastructure, and information needed to successfully run continuous experimentation at large-scale and integrate it with the development cycle, see Figure 2.5. Moreover, Fagerholm et al. [7] expect future research to apply continuous engineering to more use cases and domain-specific variants in order to expand the model.

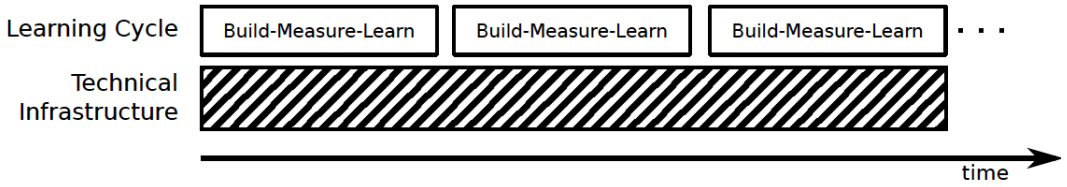

Figure 2.4: Fagerholm et al.'s [6] definition of Continuous Experimentation and its building blocks.

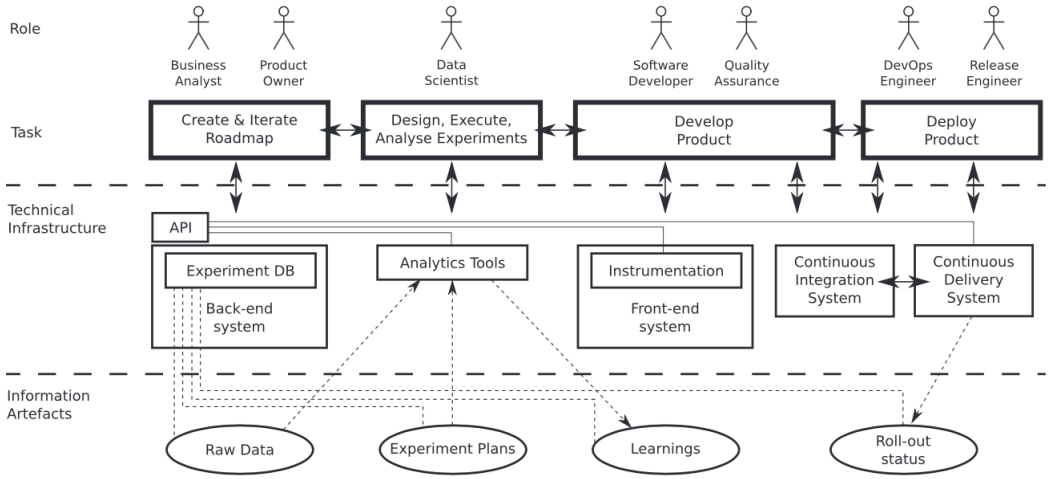

Figure 2.5: Fagerholm et al.'s [7] RIGHT infrastructure architecture for Continuous Experimentation.

2.1.5 Related Research

Rissanen et al. [25] presents a case study where they analyse the development process of two different software products in a medium-sized software company to examine whether continuous experimentation can be applied in the business-to-business domain and identify challenges. While both products were directly used by customers, i.e. external applications, Rissanen et al. anticipate three aspects of challenges: technical challenges, customer challenges and organizational challenges. The key driver for these challenges are often related to the users relying on the software, and in order to not jeopardize the user experience major changes should be avoided. Additionally, there is the organizational challenge of adapting to the experimental mindset and rely on a quantitative examination rather than opinions when making design decisions.

In another case study Kevic et al. [12] characterize the experimentation process through observation of 21 220 online experiments conducted in different environments at Bing — Microsoft's search engine. They discover that experiments can slow down the deployment cycle and therefore argue that practice should focus on identifying which experiments are worth running and making sure they run smoothly. In addition it is concluded that small code changes are usually linked to some bug fixes and are not likely to have a detectable impact on the user behaviour. Hence they imply that the cost of a controlled experiment can be lowered and the efficiency increased by tailoring experiments to the various code changes, i.e. not phrasing a hypothesis for every little code change. Even though this study issue the controlled experimentation method in a large-scale and mature product, these findings would reasonably be further applicable in the context of internal tools.

2.1.6 Challenges and Criticism

The quantitative and data driven approach to the design process has, however, risen some concerns. Incorporation of A/B testing into software development processes can evoke cultural challenges and resistance. Concerns often comes from developers that carry a sense of

ownership for the product and struggle to rely on data and launch whatever design performs best [13] (pp. 150). There is a risk of overseeing the value of the developer's experience and intuition and steer blind on data, which only gives answer through selected metrics from the hypothesis. To avoid reducing design evaluation to numbers solely Maliwat stipulates that data analytics should be balanced with alternative inputs. Data gathered from A/B testing in this study will therefore be complemented by interviews and questionnaires with relevant stakeholders. [13] (pp. 151).

Moreover there are common pitfalls to be aware of in A/B testing. Kohavi and Longbotham [3] identifies seven common pitfalls when conducting online controlled experimentation, e.g. picking metrics that hone in on a very small component of the overall objective, making it inapplicable in subsequent experiments. Another common pitfall is to forget about other differences that might occur in the treatment and not taking into account their effect on the user behaviour.

Although A/B testing brings insight into the real-world user experience and is an efficient method in decision making, it comes at a cost. Designing hypothesis and test cells, test, implement and roll them out to the entire user base take a considerable amount of time that the company of concern pays for. The consumers on the other hand will pay the in inconvenience when learning and adapting to the new changes [13].

The Effect of Long-term user learning

Encountering new design elements may cause confusion for the user on how to efficiently interact with the UI, but with time the user gradually adapts which is often referred to as user learning. Likewise, when adding new element such as a website add, users may eventually learn to ignore it. Thus when observing the user behavior in the treatment, the short-term effect is not necessarily representative of the long-term effect. Hohnhold et al. [9] present methodologies to quantify user learning in order to predict the long-term effect with only metrics measured in the short-term. Among other methods the Cookie-Cookie-Day Method (CCD) is promoted. It consists of a comparison between two parallel experiments: cookie experiment and cookie-day experiment. In the cookie each user receives the treatment every day, and in the cookie-day users initially receives the control. Subsequently, each day a fraction of the cookie-day users are randomly assigned the treatment. The cookie-day users should only be exposed on all other days, they receive the control treatment.

Essentially the users in the cookie experiment experience user learning, whereas the users in the cookie-day experiment do not receive consistent enough exposure to the treatment to obtain user learning. The learning impact on a day can thereby be derived from equation 2.3, where $\Delta M(E, C_D, D)$ is the learning impact on a day, M is the metric in question, E is the cookie experiment, E_D is current treatment in the cookie day experiment, and D is the day of choice. Calculating the user leaning on a daily basis forms a time series describing learning over time which provides an estimate of how well the test results reflects on the long term impact. The long term impact can be calculated from short-term data by subtracting the learning effect from the short-term result, see equation 2.4. E is the cookie experiment and C is the cookie-day experiment.

$$\Delta M(E, C_D, D) = \frac{M(E) - M(E_D)}{M(E)} \tag{2.3}$$

$$LT = \Delta M(E, C) - \Delta M(E, C_D, D) \tag{2.4}$$

In this study, instead of quentifying the user learning effect with Hohnhold et al's formulas, it is settled with determining whether there is a user learning effect present or not. This is done deploying an Augmented Dickey Fuller(ADF) test which is a unit root test used for determining stationarity in time series data, that is when the mean and variance are constant over time. In an ADF test a distributed lag is added to an autoregressive process model that is fitted to the time series, see equation 2.5. α, β and γ are the parameters that are to be optimized, ρ is the order of the autoregressive model, and u_t is a white noise process.

$$\Delta y_t = \alpha y_{t-1} + \sum_{j=1}^{\rho} \beta_j \Delta y_{t-j} + \gamma t + u_t \tag{2.5}$$

The solution favored by Dickey and Fuller [4] is to try the null hypothesis of a unit root, i.e. $\alpha = 0$ which implies the presence of non-stationarity. If the t-test suggests rejection of the null hypothesis there is on the other hand no sign of non-stationarity or in this study a user learning effect.

To make the costs mentioned in this Section inferior to the benefit, the MDE has to be set at a justifying level [13]. Despite the costs, Kohavi et al. [14] conclude A/B testing to have a net positive utility.

2.2 Customer Admin

Customer Admin (CA) is a tool with approximately 34 500 users where IKEA co-workers, either in the Customer Support Center (CSC) or on the store floor, can interact with customer data. CA consists of an API and a UI and has been created by IKEA Digital's Customer Engagement Team, who are currently in the process of centralizing the storage of customer data from premises to a centralized cloud service. The purpose of this Section is to explore CA's system design, create an understanding of what data is currently accessible and how it is imagined to function for its users. Although the stored information about business customers and private customers are similar, CA treats them in two different repositories. This project will only target the part of CA managing private customers, named Client. Hence this Section will exclusively present the design of CA Client. Furthermore, since there are small variations in design and accessible data across markets, this Section will focus on the Swedish market and exclude all other markets. Knowledge from this Section will guide the succeeding definition of goals and metrics in chapter 3.

2.2.1 Features

Customer Admin allow the co-worker to search for a customer in a specific market by first name, last name, e-mail, IKEA loyalty card number, city of residence, phone number, mobile number, postal code and Id. The search results show up in below the search field. Clicking on one of the results leads the user to that customer's landing page, consisting of a left side bar and a center Section.

There are different types of users in customer admin: reader, editor and admin. Readers can only see available information about a customer, editors can edit customer details and admin can do all of the above and delete customers. The left side bar displays the customer name, and has drop downs where the user can choose to see more information:

- **Contact information**, that is e-mail, phone number, mobile and address.
- **Extended Contact Information**, that is customer number, profile type, loyalty program and loyalty card number.
- **Extract Customer Information**, where the user can choose to create and download a document with all available customer information. This feature is a response to the European Union's General Data Protection Regulation.
- **Delete customer**, where a user with an Admin role has the opportunity to erase the customer from the system along with all its respective data.

The center section lays on a lightbox item and contains a top menu with five navigation buttons:

- **Overview** which is the default location. Here the user can see the customer's profile, the date of its creation and latest update. There are also editable fields containing the customer's first name, last name, social security number, customer master id, gender, date of birth and preferred store.
- **Address** containing two sections for billing address and delivery address respectively, each with editable field on the customer's street address, C/O, city, postal code and country.
- **Contact** containing editable fields with phone number and mobile number.
- **Individual** which has customer identifiers, information about consents to advertisement displaying and the customer's interest areas.
- **Transactions** which contains a sub menu with tabs to rewards details, interactions details that is about promotion, events, transaction interactions etc., purchase and order history. Under these four tabs there is no editable customer information.

When editing or adding any field under Overview, Address, Contact, Individual or Transaction, the field in question change colour from monochrome to orange. In order to save the change the user have to press the blue save button in the upper right corner triggering a pop-up asking the user to verify that the changes should be saved. When pressing "save changes" the change is saved and the field goes back to monochrome.

Below the left side bar there is also a FAQ tab that opens a FAQ compilation in the mid window.

2.3 Design Science in Software Engineering

In this research project the purpose is to achieve some research contribution and create applicable knowledge for professionals by assessing the appropriateness of A/B testing in the internal domain. Research by Engström et al. [5] signifies that design science, a common research paradigm, can be used as a lens to emphasize the scientific contribution in software engineering as applied science.

From analysing papers published at the International Conference on Software Engineering, they came up with five categories of how design knowledge is created and communicated, mapped them to the problem/solution and theory/practice domains and compiled five types of design science contributions: Problem-solution pair, Solution validation, Solution design, Descriptive and Meta, see Figure 2.6. Using the map as a lens, this project would be recognized as a solution validation since it focuses on A/B testing as a well known solution instance and validates it on the internal domain. Earlier in this chapter it has been acknowledged that previous research validates related solutions, e.g. controlled experimentation and continuous experimentation, on issued domains. Thereupon this project aims to further expand the domain specific knowledge within controlled experimentation through a proof of concept demonstrated on CA.

In preceding research, Storey et al. [29] develop a template for disclosing three aspects of design science: a technological rule encapsulating the main take away, the problem-solution pair that the research issue with an evaluation cycle, and an assessment of what value the research produce. In the next chapter, the visual abstract template is deployed in order to communicate and justify research contribution behind validating A/B testing through an implementation on Customer Admin, see Figure 3.1.

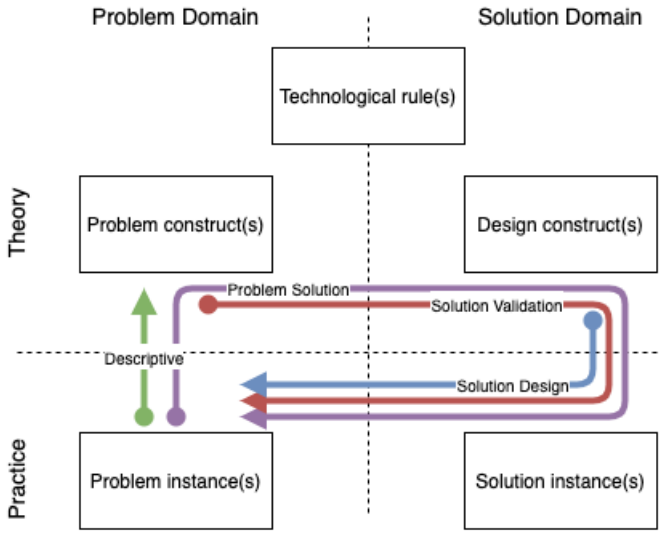

Figure 2.6: A framework for categorizing design science contributions into the problem/solution and the practice/theory domain respectively by Engström et al. [5].

Chapter 3
Research Approach

This chapter presents the activities performed when validating A/B testing in the internal domain through an implementation on CA. By way of introduction the scientific contribution is illustrated through the visual abstract template. Thereafter the implementation of A/B testing on CA broken down into definition phase, execution phase and analysis phase [13]. These phases are represented in the following chapter as Section 3.2, 3.3, 3.4. Finally the complementary questionnaires are outlined in section 3.5 and the chosen approach is motivated in Section 3.6.

3.1 Visual Abstract for Solution Validation

The research question will be answered through an implementation in CA complemented with stakeholder interviews and user questionnaires. The purpose of the implementation is to provide a practical example and a source for discussion. The aim of the interviews is to build an understanding of the overall objectives of CA, ensuring the hypothesis address a relevant issue and justifies the A/B test as a fair validation of controlled experimentation. The idea behind the questionnaires is to compare the rest result with actual user opinions and examine the experimentation's qualification to, besides the optimization objective, also reflect user preferences. See Figure 3.1 for full description of the research contribution.

3.2 Defining Goals and Metrics

The definition phase was executed through interviews with relevant stakeholders and coordinate with the development processes around CA, wherefrom goals and finally a hypothesis was extracted. The hypothesis was built in accordance with King et al's proposed framework: *We predict that [change] for [user group(s)] will achieve [effect] because of [rationale]. We will know this is true when we see [measure].* Thus all five elements were defined in this phase.

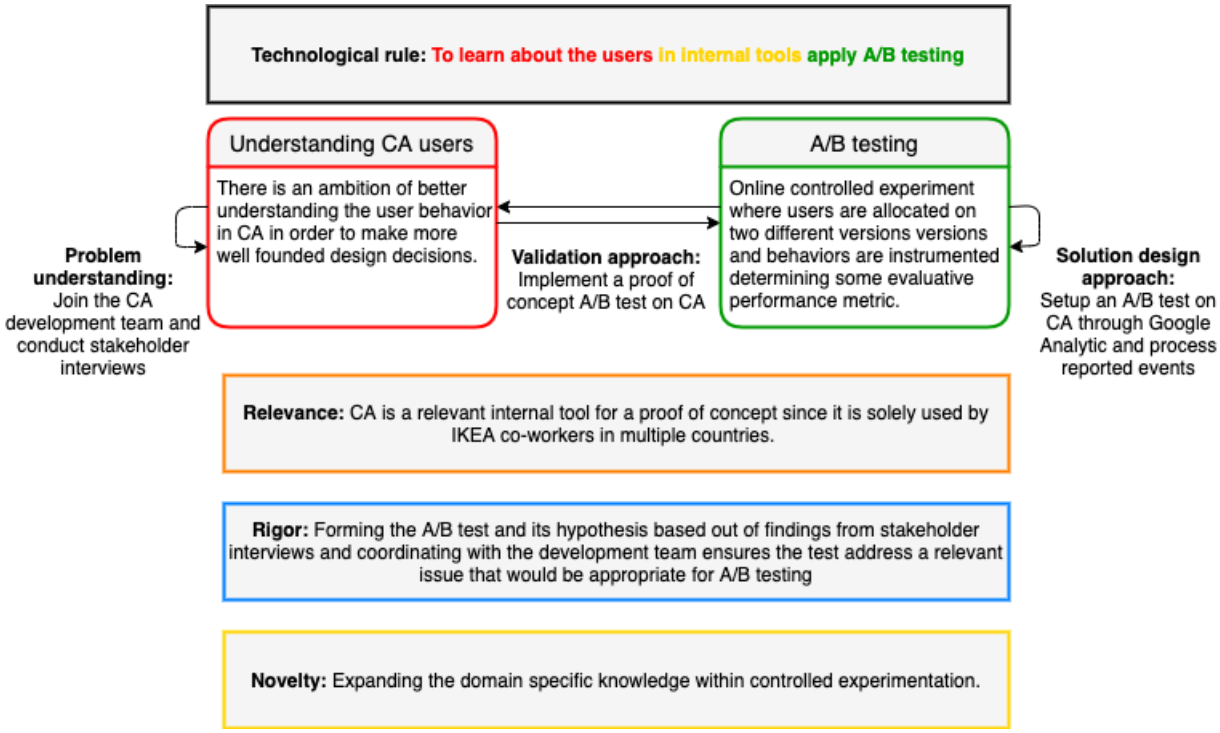

Figure 3.1: The visual abstract on the A/B testing solution validation inspired by by Storey et al. [29].

3.2.1 Interviews

As a complement to the numerical analysis and as a cornerstone in understanding the overall objectives and distill metrics in CA, interviews were conducted. The collection of interviewees can be randomly or selectively picked. Since the intention is to bring a qualitative depth rather than a support for general conclusions, focus should be kept on having a selection that covers the variation in the population [10](pp. 89–90). Moreover, taking Melegati et al's reasoning about time efficiency into account it would not have been sufficient to interview a large number of users. Therefore interviews were held with two Data Management Leader, a Process Specialist Sales Coordinator, a Database Specialist, an Engineering Manager and a Loyalty Leader, all with a connection to CA, either as a user or developer. Interviews have different levels of structure: open, semi-structured and structured [10].

The aim of the interviews was to end up with a qualitatively backed up goal definition and metrics with respect to what information was available. Therefore the interviews were semi-structured as questions were adapted and added on the fly to the respective interviewee's knowledge. See appendix A for a full compilation of the questions upon which the interviews were based. The interviews were built on four phases: context, introducing questions, main questions and a summary, and the questions were inspired by King et al. [13] addressing why and how CA is used, what stakeholders are content or discontent with and potential possibilities for improvement. Note how the main questions starts with questions that involves the refinement of goals followed by questions that involves the definition of goals. This is contradicting to the order recommended by King et al. which otherwise will be followed in this report. However, it was considered to be a more natural order in an interview, allowing

the interviewee to narrow the goals and metrics of interest through a contextual discussion.

3.2.2 Framework for Interview Analysis

After the interviews where conducted, qualitative data analysis where performed. Höst, Regnell and Runeson [10] (pp. 115) describe the qualitative analysis through four steps: data collection, coding, grouping and conclusions. One fundamental challenge in inductive research, that is research that aims at developing new theories, is that the exploratory freedom often conflicts with the high standards of rigor in scientific journals. Gioia, Corley and Hamilton [8] criticise the traditional approach to inductive concept development for being rooted too strongly in existing knowledge which delimit what can be found, and devise a systematic inductive approach that was applied to analyze the interviews. The approach can be decomposed into three steps:

- **1st Order Concepts** where the informants', i.e. the interviewees', answers and statements are compiled to a large number of informant terms forming a superficial categorization.
- **2nd Order Themes** where similarities and differences are sought among the 1st Order Concepts to condense them into a more manageable number of categories, each with a phrasal description capturing the category and preserving the informant terms.
- **Aggregate Dimensions** which is about condensing the 2nd Order Themes even further into overarching statements.

The structure of this approach enables graphically visualizing the progression from raw data to emergence of new concepts, which helps demonstrating rigor in the inductive research.

3.2.3 Adapting to the CA Development Team

In the definition phase, understanding the development team's processes and current focus was just as substantial as the interviews. Especially for the two first elements, change and user group, the definitions was inspired from a workshop, conversations and stand-up meetings with the development team. This built an awareness of what changes was on the table at the time and where an A/B test would add greater value.

3.3 Test setup in Customer Admin

This Section represents the execution phase, which is where the A/B test was executed by launching two versions of CA: one with the original design and one with a change. The test was monitored using Google Analytics(GA), a web analytics service that denotes tracking website traffic in realtime. GA was configured by creating a GA account, a Universal Analytics property with a Reporting view tracking the website URL to CA and tracking code for the property was added into the source code. Since CA is written in React, the JavaScript module react-ga was used to initialize the tracking by inserting the property's tracking ID.

A Universal Analytics property was chosen instead of a Google Analytics 4(GA4) property because of limitations in GA4's custom dimensions. In GA4, custom metrics and dimensions can only be scoped to events and users, whereas Universal Analytics provides a wide offer of scopes, including sessions. In this project, scoping custom dimensions, such as sessionID to sessions, was needed in order to group events within their session and calculate session associated metrics. Nevertheless, due to GA's processing latency Universal Analytics may cause a lower count of session. In the Google Analytics 4 model, events are processed if they arrive within 72 hours, whereas Universal Analytics only process events within 4 hours arrival time. This could have an affect on the significance of the result due to poor data volumes. Since session-scoped custom dimensions and metrics are on the product roadmap for GA4, this property type might be a better fit in the near future [16].

3.3.1 Test Cells

Two test cells where launched: Control, see Figure 3.2, which displayed the current version of CA, and Treatment, see Figure 3.3, which included a different menu design in the center section. Instead of having the five navigation buttons in a top menu and Overview as the default tab, the new menu displayed six navigation buttons including a link to the FAQ page, in squares filling out the center section. The five navigation buttons are named Purchases, Rewards, Orders, Interaction and Profile containing the Overview, Address, Contact, and Individual tabs similar to the previous menu. The first four tabs lead to the same view as the Purchase, Rewards details, Order history and Interactions details tab in the sub menu under the transaction tab in the control. The new new menu was the default landing page for the treatment group.

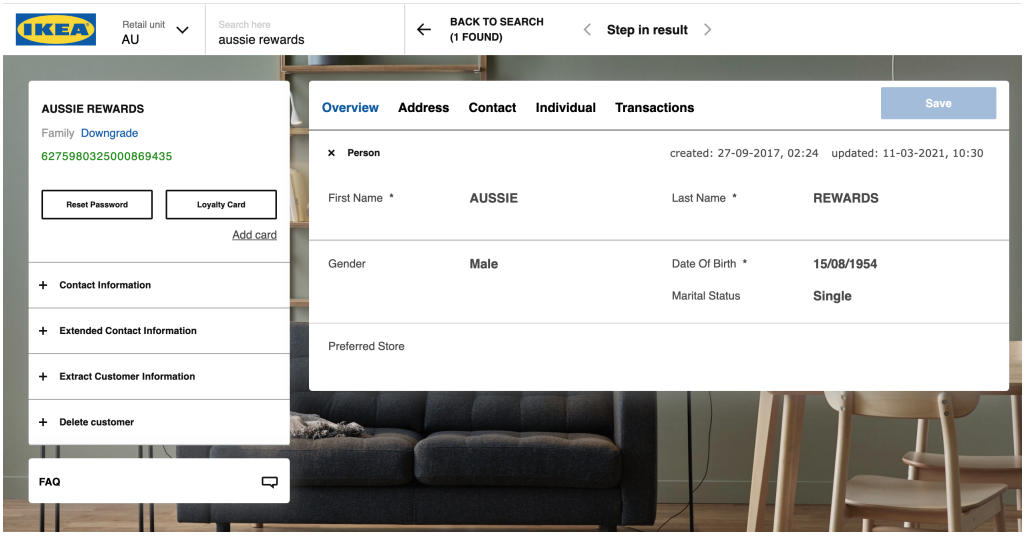

Figure 3.2: The original menu design assigned to the control. Aussie Rewards is a test profile and not a real customer.

The experiment was global in its characteristics as the treatment included a significantly different design than the control. Instead, multiple metrics was measured to build a nuanced understanding of the user behavior.

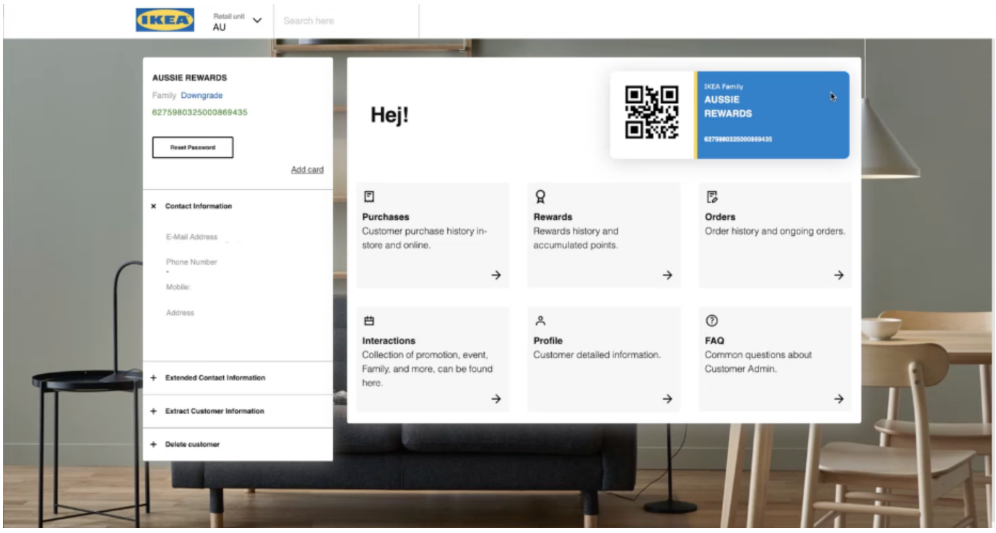

Figure 3.3: The test menu design assigned to the treatment. Aussie Rewards is a test profile and not a real customer.

3.3.2 Visitor Allocation

For each visit, users of CA Client from the Australian market were randomly allocated on the test cells, 50% on the control and the treatment group respectively. The selection of market was made under agreement with product management and development team members. Even through a larger user group would create more generous amounts of data, it was considered more cautious to conduct the proof of concept test on a small scale. In order to uphold organizational alliance, the Data Management Leader in Australia was informed about the test and the new menu design and communicated the launch to all Australian co-workers.

3.3.3 Data Collection

Data was collected for 33 days and metrics were calculated, see table 3.1.

Table 3.1: The metrics collected during the A/B test categorized into Metrics of Interest, Key Metrics and Secondary Metrics.

Metric Type	Metric Name	Explanation
Metric of Interest	Time to save change	The time it takes from entering a profile to saving a change in some customer detail within that profile in one errand.
Key Metrics	Saved Changes	How many times the "Save Changes"-button is used in an errand.
	Time per Errand	Time spent in the same customer profile when more than one click has been made in that profile.
	Clicks per Errand	How many clicks the user makes in an errand.
Secondary Metrics	Overview Clicks	How many times the Overview tab is used during an errand.
	% Overview Exits	The percentage of changes that are saved under the Overview tab.
	Clicks in Overview	How many clicks the user makes under the Overview tab in one errand.
	Address Clicks	How many times the Address tab is used during an errand.
	% Address Exits	The percentage of changes that are saved under the Address tab.
	Clicks in Address	How many clicks the user makes under the Address tab in one errand.
	Contact Clicks	How many times the Contact tab is used during an errand.
	% Contact Exits	The percentage of changes that are saved under the Contact tab.
	Clicks in Contact	How many clicks the user makes under the Contact tab in one errand.
	Individual Clicks	How many times the Individual tab is used during an errand.
	% Individual Exits	The percentage of changes that are saved under the Individual tab.
	Clicks in Individual	How many clicks the user makes under the Individual tab in one errand.
	Transactions Clicks	How many times the Transactions tab is used during an errand.
	% Transactions Exits	The percentage of changes that are saved under the Transactions tab.
	Clicks in Transactions	How many clicks the user makes under the Transactions tab in one errand.
	Purchases Clicks	How many times the Purchases tab is used during an errand.
	Clicks in Purchases	How many clicks the user makes under the Purchases tab in one errand.
	Rewards Clicks	How many times the Rewards tab is used during an errand.
	Clicks in Rewards	How many clicks the user makes under the Rewards tab in one errand.
	Orders Clicks	How many times the Orders tab is used during an errand.
	Clicks in Orders	How many clicks the user makes under the Orders tab in one errand.
	Interactions Clicks	How many times the Interactions tab is used during an errand.

	Clicks in Interactions	How many clicks the user makes under the Interactions tab in one errand.
	Profile Clicks	How many times the Profile tab is used during an errand.
	Clicks in Profile	How many clicks the user makes under the Profile tab in one errand.

KPIs such as page views, average time spent on page and bounce rate were all automatically tracked and displayed in GA. Clicks on specific buttons and timestamps of those clicks, however, was not available and had to be customized through 14 different tags in Google Tag Manager(GTM), see list below. Similar to the tracking configuration in GA, a tracking code was added into the source code using react-gtm. In GTM, each tag was associated with a trigger that was activated on a specific button. Once a tag was triggered, an event was created in GA.

- **Enter Profile**, triggered whenever a user clicks on a customer profile from the search results.
- **Save Changes**, triggered when a user clicks on the confirmation button with click text "Save Changes".
- **Reset Password**, triggered when a user clicks on the button with click text "Reset Password".
- **Overview Clicks**, triggered when a user clicks on the Overview navigation button.
- **Address Clicks**, triggered when a user clicks on the Address navigation button.
- **Contact Clicks**, triggered when a user clicks on the Contact navigation button.
- **Individual Clicks**, triggered when a user clicks on the Individual navigation button.
- **Transaction Clicks**, triggered when a user clicks on the Transaction navigation button.
- **Purchases Clicks**, triggered when a user clicks on the Purchases navigation button in the Test version.
- **Rewards Clicks**, triggered when a user clicks on the Rewards navigation button in the Test version.
- **Orders Clicks**, triggered when a user clicks on the Orders navigation button in the Test version.
- **Interactions Clicks**, triggered when a user clicks on the Interactions navigation button in the Test version.
- **Profile Clicks**, triggered when a user clicks on the Profile navigation button in the Test version.
- **Non Key Clicks**, triggered by any click that is not triggering any of the clicks above.

In order to comply with GDPR, IP addresses were made anonymous in the tag configuration. Since time to save a change was conditional on two events, profiles entered and saved changes, it could not be displayed directly in GA. For each session the time differences between those two events were calculated in a Python script, see Section 3.4. Moreover, four custom JavaScript variables were created and sent in custom metrics or custom dimensions along with each event:

- **Timestamp** — the unix time of when a tag is triggered. This variable was sent in a custom metric with each event.
- **SessionID** — a unique identifier for each session. This variable was sent in a custom dimension with each event.
- **EventID** — a unique identifier for each event. This variable was sent in a custom dimension with each event.
- **FullURL** — the URL to the page where the event was triggered. This variable was sent in a custom dimension with each event.

Because of relatively low traffic, data was collected longitudinally for 33 days.

3.4 Data Analysis

In the analysis phase the collected data from the execution phase was used to reject or accept the hypothesis. All data collected from the triggered events where listed in a google sheet by Google Analytics Spreadsheet Add-on and downloaded as a csv-file. The csv file was processed in a Python script that grouped all events in their session. In each session errands were extracted, containing all events triggered between entering and leaving a customer profile. Whenever no clicks had been made between entering and leaving a customer profile that errand was discarded in the statistics under the assumption that the user had entered wrong customer profile. The script sorted the errands into control and treatment by initially assigning all errands to the control and then moving errands to treatment whenever a test-specific event was triggered in that errand's session. Each metric in table 3.1 was calculated per errand and 90% confidence intervals were calculated for respective metric. Welch's t-test was performed on the metric of interest and compared with critical value for the t-test for significance level of 10%. This relatively low level was selected due to low traffic, which gave reasons to presume small data volumes and therefore low statistical significance.

The data processing script was written and used in advance to the real A/B test when all users were still using the original menu. This dry run facilitated testing the script, detecting bugs and other systematic failures. Moreover the standard deviation of respective metric was derived, wherefrom the minimum sample size was calculated through equation 2.2.

The extraction of errands was, however, not the initial approach. Originally the metrics in table 3.1 were calculated once per session, but data collected from the dry run revealed that sometimes the user saved changes in multiple customer profiles in the same session. In Universal Analytics a session starts by default, when a user enters the website and stops either when there is 30 minutes of inactivity or at midnight [17]. Since IKEA co-workers might manage several customer errands using CA without being inactive for 30 minutes, extracting

errands in the python script would allow for larger amounts of data to be collected. An errand was defined so that it should include all events that a user triggered while handling the same customer. Therefore the "Enter Profile" was used as an indicator of that a new customer errand was being encountered. Whenever a change was saved or a new profile was entered the previous errand ended.

Additionally, in order to explore the presence of a user learning effect, all time differences were mapped with the date of the session and plotted as a time series. An augmented Dickey-Fuller test was then performed on that time series to determine stationary, that is when a time series appear to have a constant mean, standard deviation and co-variance.

3.5 Post Collection Questionnaires

As a complement to the data collection, a questionnaire was sent out and answered by 8 users, see appendix C. The purpose of gathering actual user opinions was to compare it with the test result and examine the experimentation's ability to, besides the optimization objective, also reflect user preferences. Since CA is a part of the co-workers working environment that IKEA provide, personal comfort and a sense of well-being is of corporate interest. Therefore the questionnaire included questions addressing aspects such as personal experience, aesthetics and user friendliness. The questionnaire should be considered semi-open since there were open-ended questions associated with closed-ended ones giving the user a chance to rate both the variants and the overall experience on these aspects.

3.6 Motivation of Approach

The Customer Admin team has been looking for ways to make more data driven design decisions awhile. Since they started developing the tool design decisions have been based out of user stories and UX Designers' believes in how features will enhance the user experience. Therefore the proof of concept implementation did not only serve as a practical case and a source of discussion on the research question, but even so as a demonstration for the development team on how they can apply controlled experimentation to access real world user feedback.

In executing this method, in accordance to King et al's [13] three phases and Fagerholm et al's [6] build-measure-learn block, the outcome of one step, e.g. the hypothesis definition, impacts the characteristics of subsequent steps. For that reason it was insufficient to explicitly outline the full approach in advance for this project. Instead, it should be considered as an agile process, where the execution and analysis approach were continuously refined as previous steps where carried through.

In addition, this A/B test was a first experiment iteration in the development cycle. Therefore it the efforts behind the stakeholder interviews was prominent as it was critical to accurately map the objectives and stakeholder interests in order to effectively start the cycle of hypothesis engineering. Afterwards when A/B testing is adapted as a continuous experimentation, findings from previous tests will guide and help fuelling the next hypothesis.

Chapter 4
Results in Relation to Research Question

In this chapter, data gathered from the interviews are presented in first order concepts, second order themes and aggregate dimensions according to Gioia et al's [8] framework in Section 4.1. The data gathered from the A/B test are presented for respective metric in Section 4.2, on which the hypothesis is tested in Section 4.3. Lastly, the answers received to the questionnaire are compiled in Section 4.4.

4.1 Goals and Metrics

The results from the definition phase are a compilation of the interview findings and reasoning with the development team into a hypothesis. First order concept that is quotes from the interviewees, second order themes and aggregate dimensions are shown in Figure 4.1, 4.2, 4.3, 4.4 and 4.5 accordingly with Gioia et al's three-step-model [8].

In Figure 4.1, informant statements addressing the overall objective and purpose of CA are compiled in 1st order concepts. In the second order themes, three main components were derived from these informant statements: making customer data visible and manageable for the co-worker, providing the co-worker with an instrument to manage customer requests through, and that it should support the co-workers' time efficiency. Lastly, these components are summed in one aggregate dimension.

In Figure 4.2, informant statements describing how CA is used are compiled in 1st order concepts. As second order themes, six components were derived acknowledging that CA is used by co-workers mostly in CSCs to accomodate incoming customer requests. After searching for– and selecting a customer, the co-worker asks control questions on information displayed in the profile in order to make sure they've selected the right customer. Thereafter two different use cases are identified: 1. Looking up data for the customer, such as transactional data for customers that have lost their receipts, 2. Helping the customer to update some information, such as email, telephone number or resetting the password. The six components were aggregated into one single dimension.

In Figure 4.3, informant statements that explained how the usability is perceived and how it can be measured were extracted from the interview transcripts. Four 2nd order themes were derived and aggregated into two different dimensions: 1. Prioritizing the displaying of information makes it easier for the user to complete their task, 2. The "Save Changes" button is a good indication of that a user has finished a task but not an insurance it was correctly completed.

In Figure 4.4 challenges and problem areas identified by the informants were compiled in 1st order concepts and merged into three 2nd order themes explaining that a customer errand can take about 6 minutes, but that there are additional time consuming factors such as chatty customers or scarce information. Moreover there was a newly launched restriction against making duplicate customer profiles which had caused some confusion among users.

Figure 4.5 compiles informant quotes that shed light on potential opportunity areas and features that could be added to CA. Note that this Figure only displays 1st order concepts and 2nd order themes. This is because the interviewee quotes were concise and sufficient enough to present in second order themes.

The key findings suggests that the goal of CA is to assist the co-workers in accomodating incoming customer requests that regards either lookup or update of some customer profile data, which are also the two most frequent use cases. Making the tool intuitive by showing the most important information first would enhance the usability and hence probably shorten the time spent per customer errand.

With the aggregate dimensions an hypothesis was adapted to the development teams current visions:

We predict that a new start menu for co-workers in the Australian market will shorten the average time spent per errand because it makes the orientation in CA easier to comprehend. We will know this is true when we see a decreasing time difference from entering a profile and saving a change.

The errand of updating a customer data was chosen instead of customer data lookup due to the clear indicator when saving a change, see Figure 4.3. Finding the stop time for when the user has found the sought information would not enable the same precision.

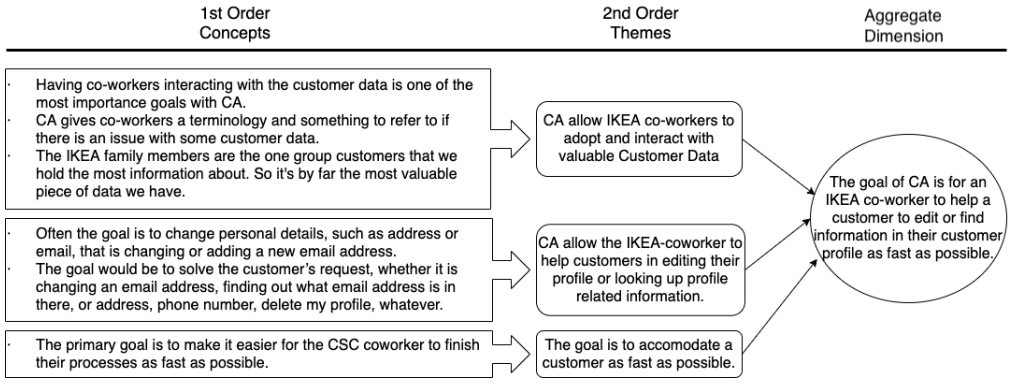

Figure 4.1: First order concepts, second order themes and aggregate dimension of the overall goal of CA. [8].

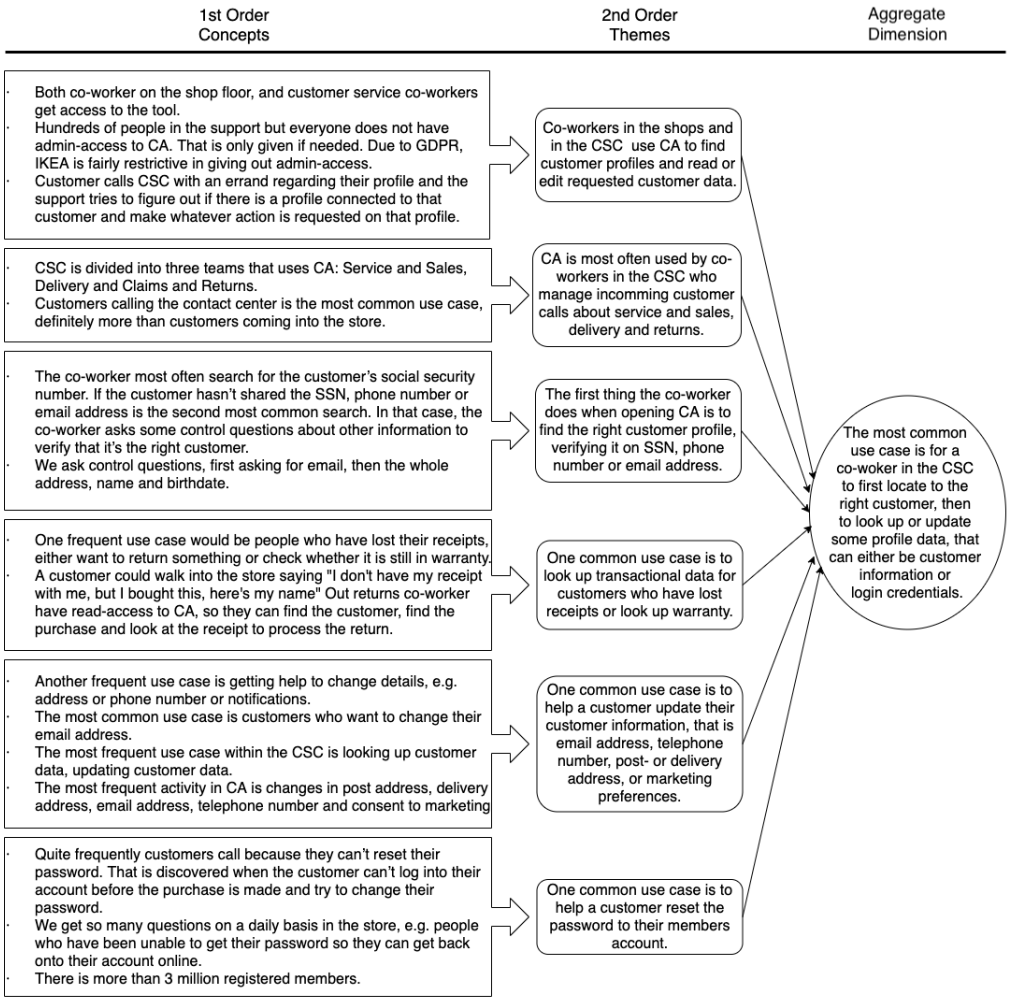

Figure 4.2: First order concepts, second order themes and aggregate dimension of how CA is used. [8].

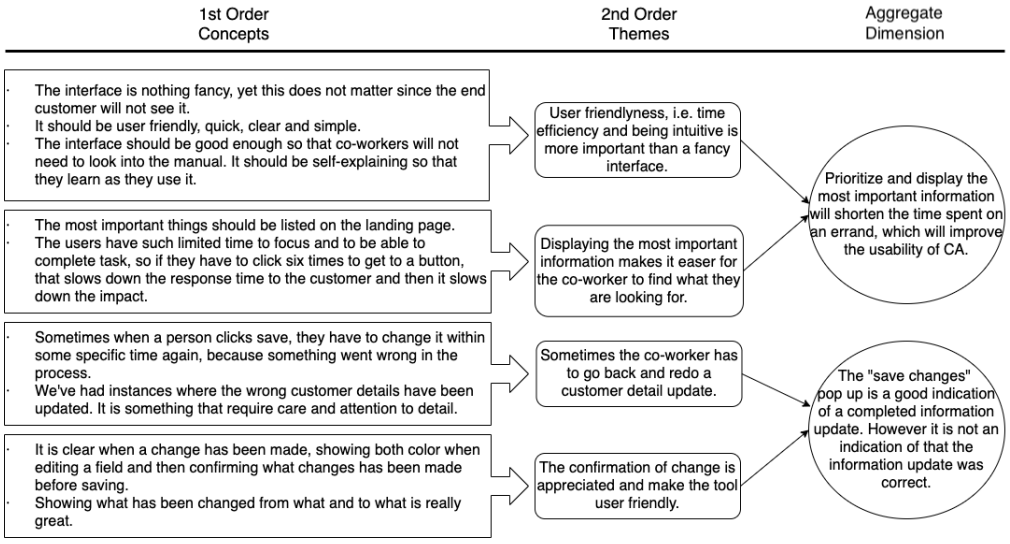

Figure 4.3: First order concepts, second order themes and aggregate dimensions of what makes CA userfriendly. [8].

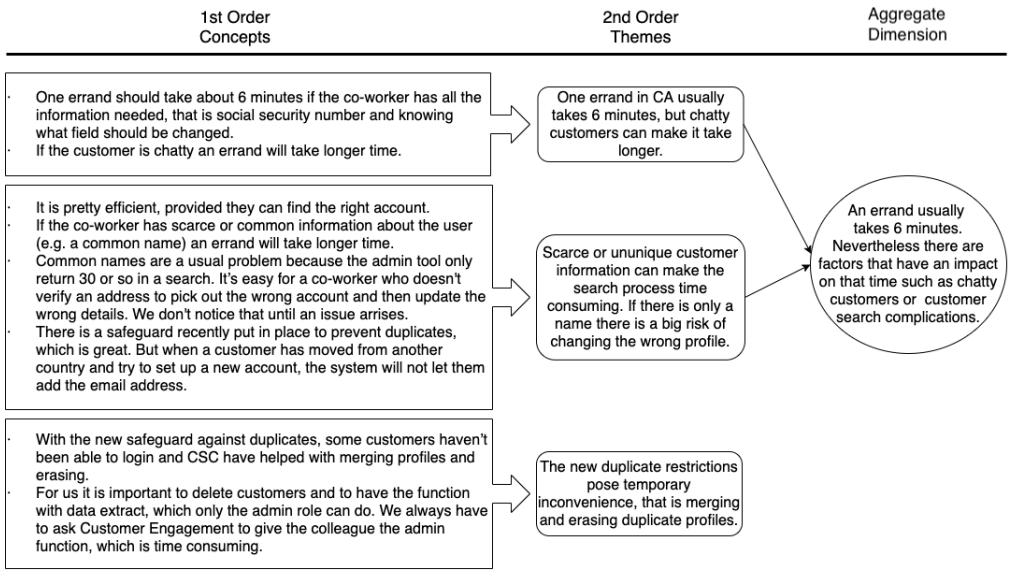

Figure 4.4: First order concepts, second order themes and aggregate dimensions of common issues and time allocation in CA. [8].

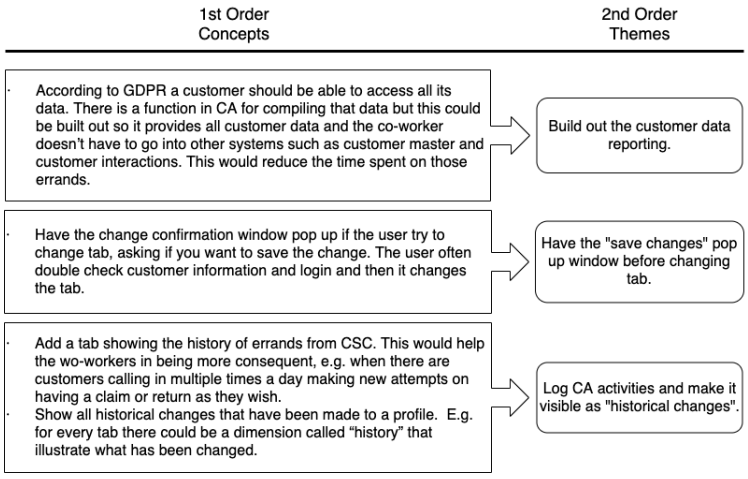

Figure 4.5: First order concepts and second order themes of opportunity areas for CA. [8].

4.2 Data and Statistics

An A/A test was executed 5 days in advance to the A/B test. Instead of allocating users on two different versions, the A/A test assigned users to one single version, usually the original version. This can be conducted in prior to the A/B test to try the experimentation system and detect flaws.

From the collected data and equation 2.2 the minimum sample size was calculated to 1568 errands, with $\sigma = 99$ and $\Delta = 10sec$. On average 12 out of 437 errands collected per day had a change that had been saved. Hence it would take approximately $\frac{1568}{12} = 130$ days to collect the minimum sample size. Even though this was considered to long and would exceed the duration of the project, the A/B test was launched and data was collected for 35 days.

Data gathered from the control and the treatment are presented in table 4.1. The metrics that got significantly differing results on a 90% confidence level are marked in bold. The ADF tests gave 0 in p-value for both the control and the treatment and hence there is no sign of non-stationarity that would indicate on a user learning effect being present.

Table 4.1: Mean values and 90% confidence intervals on the metrics gathered from the A/B test.

Metric Type	Metric Name	Mean Control	Mean Treatment	p-value
Metric of Interest	Time to save change (sec)	89.8 ± 14.0	90.4 ± 31.4	0.488
Key Metrics	Saved Changes	0.0289 ± 0.0032	0.0339 ± 0.0080	0.167
	Time per Errand (sec)	232 ± 8	186 ± 15	0.099
	Clicks per Errand	8.33 ± 0.19	9.03 ± 0.39	0.00418
Secondary Metrics	**Overview Clicks**	0.100 ± 0.009	0.134 ± 0.023	0.0135
	% Overview Exits	0.399 ± 0.009	0.0440 ± 0.0090	1
	Clicks in Overview	2.16 ± 0.09	0.659 ± 0.110	1
	Address Clicks	0.138 ± 0.011	0.178 ± 0.029	0.0160
	% Address Exits	0.0179 ± 0.0025	0.0130 ± 0.0050	0.927
	Clicks in Address	0.186 ± 0.062	0.114 ± 0.043	0.943
	Contact Clicks	0.179 ± 0.012	0.229 ± 0.032	0.00754
	% Contact Exits	0.0476 ± 0.0040	0.0707 ± 0.0113	0.000801
	Clicks in Contact	0.350 ± 0.038	0.506 ± 0.102	0.00949
	Individual Clicks	0.202 ± 0.011	0.234 ± 0.029	0.0489
	% Individual Exits	0.0368 ± 0.0036	0.0339 ± 0.0080	0.707
	Clicks in Individual	0.276 ± 0.033	0.335 ± 0.078	0.128
	Transactions Clicks	0.440 ± 0.013	0.190 ± 0.024	1
	% Transactions Exits	0.499 ± 0.010	0.141 ± 0.015	1
	Clicks in Transactions	3.04 ± 0.14	0.978 ± 0.133	1
	Purchases Clicks	0 ± 0	0.330 ± 0.026	0
	Clicks in Purchases	0 ± 0	2.91 ± 0.26	0
	Rewards Clicks	0 ± 0	0.051 ± 0.010	0
	Clicks in Rewards	0 ± 0	0.201 ± 0.094	0.000211
	Orders Clicks	0 ± 0	0.0411 ± 0.0088	0
	Clicks in Orders	0 ± 0	0.190 ± 0.072	0
	Interactions Clicks	0 ± 0	0.0375 ± 0.0085	0
	Clicks in Interactions	0 ± 0	0.146 ± 0.048	0
	Profile Clicks	0 ± 0	0.149 ± 0.019	0
	Clicks in Profile	0 ± 0	0.413 ± 0.083	0

4.3 Hypothesis Evaluation

In the control, it took on average 89.8 seconds for a user to save a change in a newly entered customer profile. In the treatment, the average time difference was 90.4 seconds. Nonetheless, since the p-value of 0.488 suggest low probability of them differing we can not accept the hypothesis on this level of confidence. Hence we can not determine from the hypothesis whether the new menu design has made the users more efficient in updating customer information.

Looking into the key metrics the control had on average 0.0289 saved changes per visit, i.e. a change was saved during 2.89% of the visits. In the treatment a change was saved during

3.39% of the visits. Moreover the average time spent per errand was 232 seconds in the control and 186 seconds in the treatment, and the average number of clicks per errand was 8.33 in the control and 9.03 in the treatment.

4.4 Questionnaire Answers

In total 8 users answered the questionnaire, which is shown in Table 4.2. In summary there was no prominent dissatisfaction with the old menu design. The users' evaluation of the new menu design is, however, fairly higher and 77.8% of the questionnaire participants prefer the new menu design to the old.

Table 4.2: Mean values from the 5 point rating on the usability and the aesthetics of the control and treatment respectively. The user's version preference is shown in percentage on the third row.

Question	Mean Original	Mean Treatment
Rate the user friendliness (1-5)	3.56	3.89
Rate the aesthetics (1-5)	3.22	4.33
What version do you prefer?	22.2%	77.8%

Chapter 5
Discussion

In this chapter the insights from the proof of concept implementation are broken down and presented in four Sections. Section 5.1 presents CA related insights and findings. After the test execution was completed, the Australian market went back to the original design as the rest of the markets. The development team desired to launch it globally in the future although further refinements and briefings to concerned stakeholders was needed in prior. Section 5.2 compiles the main challenges from the proof of concept and attempts to generalize to the internal domain, Section 5.3 comprises some key takeaways for A/B testing an internal tool, and Section 5.4 presents a proposal for future research projects. It is acknowledged that the value of a user feedback loop, in terms of gained user behavior insights, is indeed present in the internal domain. Among other challenges, the relatively low traffic poses a substantial risk of receiving scarce data and ending up with poor statistics. To overcome this challenge larger test groups and more frequently triggered metrics of interest is proposed.

5.1 Customer Admin User Behavior Insights

The excessive time it takes for a user to update some customer profile in the new menu, 90.4 seconds instead of 89.8 seconds, suggests that the control menu design is preferable to the new design. There are, however, some lack of confidence due to the low numbers of errands in these statistics which can be explained through the low frequency of saved changes, 0.029 and 0.034, in CA per errand. This along with the aggregate dimension in Figure 4.2, indicates that looking up profile data is a more common use case than updating profile data. Contrarily, looking at the time difference per errand where the time stops whenever the user has left the customer profile it appears the new menu design has shortened the time spent per errand with 46 seconds. Here, the standard deviations are significantly smaller which is likely because of the inclusion of more use cases, resulting in a substantial sample size.

In the control version, the transaction tab beats the other tabs both on number of tab clicks, number of clicks under a tab, and in being the last tab before leaving the customer

profile. Since going into the transaction tab doesn't provide the user with the possibility of updating a customer profile, those errands were presumably about looking up transactional data which again advocates for that data lookup is more common than data update. Since the information under transaction and its sub menu's tabs seems important, the aggregate dimension in Figure 4.3 of prioritizing and displaying the most important information first, propagates for the new menu design were those tabs are displayed on the landing page.

5.1.1 User's Demands in Customer Admin

From the perspective of users of internal tools, i.e. co-worker, the tool is one component at their working desk. As structure and aesthetics are recognized as important aspects of a working environment, having an impact on the overall work performance, these factors should be taken into consideration as well. Moreover, as mentioned in the interviews shown in Figure 4.3, having all data that is needed available in CA is determining for CA's fulfillment of the users' needs.

5.1.2 Hypothesis Engineering in Developing Internal Tools

Incorporating hypothesis engineering into the development methodology induces fueling subsequent development ideas and hypotheses with data and user behavior insights from previous tests. Therefore, the A/B testing process and its three phases should be seen as a circular process rather than a linear, where insights from the analysis phase should be fed into the definition phase of the next A/B test as illustrated in Figure 2.2. But in order to efficiently utilize those insights to identify, specify and prioritize hypotheses there are organizational challenges found by Rissanen et al. [25] that will have to be faced. Therefore a culture of innovation within the organization should be created and new ideas promoted since it requires some level of creativity to come up with hypotheses questioning the current design. Secondly developers should be educated in controlled experimentation in order to grow competence and understanding. Finally the transition towards hypothesis engineering should be integrated with other affected teams so that the experimental mindset establishes organically and people throughout the organisation are comfortable in making data driven design decisions.

5.2 Main Challenges

This Section comprises main challenges when setting up and executing an A/B test and justifies some of them as potential cost drivers.

5.2.1 Traffic Flow and Data Volume

The low flows of traffic, that is approximately 112 collected errands per day where the user has saved a change, leads to scarce data and insignificant hypothesis evaluation. To put this

in comparison with other A/B tested websites and tools, Facebook had on average 1.85 billion daily active users in 2020 [22], Amazon had over 2.44 billion visitors in 2020 [21], and Netflix had over 207 million subscribers in the first quarter of 2021 [23]. The number of visitors on these external websites and platforms, are greatly exceeding the amount of coworkers at IKEA and potential users of CA and hence also the potential for having significant test results. Principally, there are no companies with an exceeding amount of employees — as an illustration the majority of the world's 50 largest companies have less than 400 000 employees [24]. Therefore A/B tests of internal tools will not reach the same level of reliability as A/B tests of external websites where customers are end users.

The abundant traffic in external tools allows for collecting data in a snapshot and get a enough data in only a few days, unlike when testing internal tools where it can take months to get a sufficient result. This time evokes a cost in terms of placing the test users' experiences into jeopardy during the execution phase. This comprises both having the users in a potentially poorer UI and giving the user an ambiguous experience, occasionally alternating and being adaptive between two experiences for a longer time.

5.2.2 Sources of Error in Data Collection

Apart from the insufficient amount of collected data, the quality of the data was partially lacking. As stipulated in the aggregate dimension in Figure 4.4, there are multiple factors affecting the time it takes for a co-worker to manage an incoming customer request. Chatty customers can make it take longer or appearance of any other happening that interrupts the interaction with the customer. These factors are external and cannot be mitigated or influenced by the design of the UI. Even though they do not distort the average values they increase the variance of the test result.

Secondly, as declared in Figure 4.3, the save changes are not necessarily an indication of that the user successfully finished the task of updating a customer detail. In the analysis the time to save a change is defined as the time difference between entering a profile and saving the first change. However, in case the user finds out that the customer detail was updated incorrectly, the time computing should end when the user has gone back and corrected the detail. This should not affect the A- and B version differently, yet there are reasons to believe that the calculated average time to save a change may be an underestimation for both versions.

Finally there is the uncertainty about whether the user has even located itself the right customer. As insinuated by Figure 4.2 and Figure 4.4 occasionally the user does not find the right customer at once, especially when the user is given scarce or common customer information for searching. Therefore in addition, the user asks verifying control questions when entering a profile. Nevertheless, in the analysis all profile visits where clicks had been made in the profile were counted as errands. As for the selection of errands this may cause an inaccurate inclusion of profile visits where the user discovers it is the wrong profile and leave instantly. The consequence of this is that those errands presumably take shorter time than an errand were the user lookup or update information, and thereby shorten the average errand time.

The sources of error above are solely affecting the statistics of the user behavior and hence only the testers and the developers test result. Therefore these are not considered direct cost drivers but rather causes for loss in value of clear and efficient user feedback.

IKEA's Work Process and The Network Effect

Another challenge that generates a source of error is the users' awareness of being tested. In accordance with IKEA's working process and training standards, concerned co-workers and even the users themselves had to be briefed about the new landing page before launch. Even though they were not informed of the test's purpose and how performance was measured, the users' awareness did probably affect the result.

Firstly, by telling the co-workers that they will get something new, it is plausible that they form an opinion already before using the new version. Users that like change would probably form a positive attitude whereas users who are more conservative would become sceptical. Subsequently, the predetermined attitude would expectantly affect their efficiency in respective version and thereby the test-statistics. Since it is prominent that the users themselves prefer the new version to the old, see Table 4.2, there are reasons to believe that there is a factor of preference, e.g. in the metric "Time per Errand".

Apart from the preferential factor, the user briefing likely had an impact on the user learning effect. By providing the users with instructions and help on how to use the new interface, they learned outside of their own interactions and the user learning effect should thereby not be as apparent in the treatment. This could explain why the ADF tests showed no sign of non-stationarity and thereby no signs of a user learning effect.

Moreover, since co-workers in the same CSC are gathered in the same office building and inevitably collaborate, the test users influence each other, which violates the stable unit treatment value assumption (SUTVA) [28]. Saveski et al. [28] claim that in an A/B test the treatment should only have an impact on the users being treated — not other users that are in connection with them. In alignment with Backstrom et al. [1] the co-workers' influence on each other would be described through the term "Network Effect" which creates an interdependence in the random sample, violating the assumption of independence.

5.2.3 Customizing Metrics and Extracting Single Customer Interactions

When introducing A/B testing on CA as an internal tool, it was necessary to carry out the definition phase thoroughly and outline the fundamentals and objectives of CA. In order to serve the goal in Figure 4.1 of shorten the time spent per customer interaction, the majority of the metrics of interest and the key metrics, such as time to save change; time per errand; and clicks per errand, were about minimizing time or the number of clicks between two specific events. Plausibly, this principle stands applicable for internal tools in general as the users' i.e. coworkers' time efficiency is a matter of interest from a profits perspective.

Subsequently when exploring available web analytic services and test tools, e.g. Google Optimize and Optimizely, for the execution it was prominent that the tools were adapted to the objectives of external websites and the inverted optimization problems such as maximization of pageviews, subscriptions, revenue and conversions which are common metrics for publishing websites and ecommerce websites [18]. Thus, depending on business model and what KPIs are sought, the setup in of an A/B test for an external website or app domain could be fairly straight forward if the key metric is already available in the test tool. Customized metrics could be made but were limited to the aggregation of discrete events from GTM. Since the available metrics nor the customizables did not satisfy the needs of tracking

time and clicks between two specific events, a script had to be written in order to determine the metrics of interest and key metrics. The process of setting up an A/B test for an internal tool is therefore reasonably not as straight forward and somewhat costly in relation to setting up an A/B test in the external domain, depending on business model and KPIs.

5.3 Recommendation

In this Section, the value of A/B testing internal tools is compared with the cost in order to determine the net utility and furthermore find a recommendation on how to proceed with the experimentation method addressing the challenges identified in the previous Section.

5.3.1 Expand Data Source

In order to respond to the loss in reliability due to low traffic and data insufficiency, the user group could be expanded to include more markets. The enhanced data flow would both increase the significance levels and shorten the execution time needed.

Notwithstanding there are risks in exposing a larger amount of the users to the test. If this A/B testing behavior becomes the common, the ambiguity of alternating between two UI versions will become a binding part of the overall user experience. For that reason, it would be sensible to select a more local scope for the market wide tests where there are only one or two variables in a test cell that does not create a noticeably different experience. The change in this test for example, would not suit a market wide experiment as it was global and exposed the user for a markedly new orientation model. On the other hand, a local scope would likely generate a smaller impact on the users' behavior and thereby make it even more complicated to detect significant differences.

5.3.2 Specify Insensitive Metrics

Another way to respond to the scarce data is to include more frequent use cases such as the use case where the coworker looks up some customer information, as a suggestion through the metric time per errand. As shown in table 4.1, the key metric time per errand provided a comparable measurement with non-overlapping confidence intervals. Therefore, although this proposal would imply some challenges in finding good indicators as explained in Section 4.1, it is preferable before only tracking errands where changes are saved.

As described in Section 5.2.2, time is a unit that is sensitive to disturbance or other factors that are not a part of the UI. With that in mind the number of clicks to finish an errand would be a more suitable unit to quantify the co-workers efficiency when managing a customer request. In compliance with this reasoning, the standard deviation on the number of clicks per errand shown in table 4.1 are small enough to provide non-overlapping confidence intervals. Yet awareness should be payed to that the number of clicks does not take into account the effort the user puts into finding the data field or button he or she looks for, which certainly is a factor in the coworker's efficiency.

5.3.3 Streamline Setup

In order to minimize the efforts and hence the cost of installing and executing the test, the setup should be built so that redundancy is avoided. Foremost it is about choosing the right web analytic services that efficiently track the data of interest and forward it to platforms and tools that are already deployed and used. For instance, if using GTM the triggers should be configured on CSS selectors, unlike in this test were most triggers were configured on click text since that was simpler to setup. The disadvantage is that if a click element is added in the future containing the same string, the tag will trigger on that button as well and the events will no longer be collected correctly.

Besides, the script should extract as generic metrics as possible that quantifies the goal of most UX redesigns. In this case the time spent per errand and the number of clicks could be considered usable metrics in succeeding A/B tests, and hence it is in ambition to reuse the script in the future.

5.4 Conclusion and Further Work

The proof of concept A/B test implementation on CA unveiled the experimentation as an instrument for evaluating new features in internal tools through real user feedback and that further expands the knowledge about the users. There are, however, domain specific challenges when efficiently incorporating A/B testing and the circular process of hypothesis engineering into the development of internal tools, e.g. the low traffic flows resulting in scarce data, sources of error in the data due to inadequate task completion indicators, and customization of metrics to the objectives of the tool in question. Having a greater source of test users and selecting more frequently triggered metrics of interest could mitigate the data scarcity which is therefore proposed.

Future research should focus on deepening the knowledge of how A/B testing can be implemented in the internal domain. This would preferably be done through more case studies where similar proof of concept implementations were executed on other types of internal tools, such as inventory management systems or other administration systems, adding more rigour to the solution validation. These implementations would further enhance the rigour by applying more local or global scopes and different levels of fulfillment, i.e. explorative or evaluative.

Finally future research could potentially expand the solution validation onto other domains, similar to previous research by Rissanen et al. [25]. This would extend the horizon of how to utilize the value of real world user feedback from controlled experimentation.

References

[1] Lars Backstrom and Jon Kleinberg. Network bucket testing. New York, NY, USA, 2011. Association for Computing Machinery.

[2] R. Chatley. Supporting the developer experience with production metrics. In *2019 IEEE/ACM Joint 4th International Workshop on Rapid Continuous Software Engineering and 1st International Workshop on Data-Driven Decisions, Experimentation and Evolution (RCoSE/DDrEE)*, pages 8–11, 2019.

[3] Thomas Crook, Brian Frasca, Ron Kohavi, and Roger Longbotham. Seven pitfalls to avoid when running controlled experiments on the web. New York, NY, USA, 2009. Association for Computing Machinery.

[4] David A Dickey and Wayne A Fuller. Likelihood Ratio Statistics for Autoregressive Time Series with a Unit Root. *Econometrica*, 49(4):1057–1072, June 1981.

[5] Emelie Engström, Margaret-Anne D. Storey, Per Runeson, Martin Höst, and Maria Teresa Baldassarre. A review of software engineering research from a design science perspective. *CoRR*, abs/1904.12742, 2019.

[6] Fabian Fagerholm, Alejandro Guinea, Hanna Mäenpää, and Jürgen Münch. Building blocks for continuous experimentation. 06 2014.

[7] Fabian Fagerholm, Alejandro Sanchez Guinea, Hanna Mäenpää, and Jürgen Münch. The right model for continuous experimentation. *Journal of Systems and Software*, 123:292–305, 2017.

[8] D. A. Gioia, K. G. Corley, and A. L. Hamilton. Seeking qualitative rigor in inductive research: Notes on the gioia methodology. *Organizational Research Methods*, 16(1):15–31, 2013.

[9] Henning Hohnhold, Deirdre O'Brien, and Diane Tang. Focus on the long-term: It's better for users and business. In *Proceedings 21st Conference on Knowledge Discovery and Data Mining*, Sydney, Australia, 2015.

[10] Martin Höst, Björn Regnell, and Per Runeson. *Att genomföra examensarbete*. Studentlitteratur AB, 2006.

[11] J. O. Johanssen, L. M. Reimer, and B. Bruegge. Continuous thinking aloud. In *2019 IEEE/ACM Joint 4th International Workshop on Rapid Continuous Software Engineering and 1st International Workshop on Data-Driven Decisions, Experimentation and Evolution (RCoSE/DDrEE)*, pages 12–15, 2019.

[12] Katja Kevic, Brendan Murphy, Laurie Williams, and Jennifer Beckmann. Characterizing experimentation in continuous deployment: A case study on bing. In *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP)*, pages 123–132, 2017.

[13] R King, E.F Churchill, and C Tan. *Designing with Data*. O'Reilly Media, Inc, USA, 2017.

[14] R. Kohavi, R. Longbotham, D. Sommerfield, and et al. Controlled experiments on the web: survey and practical guide. *Data Mining and Knowledge Discovery*, 18:140–181, 2009.

[15] Ron Kohavi and Roger Longbotham. *Online Controlled Experiments and A/B Testing*, pages 922–929. Springer US, Boston, MA, 2017.

[16] Google LLC. Analytics Help [ga4] custom dimensions and metrics, 2021.

[17] Google LLC. Analytics Help how a web session is defined in universal analytics, 2021.

[18] Google LLC. Optimize Resource Hub how to configure experiment objectives in optimize., 2021.

[19] J. Melegati, X. Wang, and P. Abrahamsson. Hypotheses engineering: First essential steps of experiment-driven software development. In *2019 IEEE/ACM Joint 4th International Workshop on Rapid Continuous Software Engineering and 1st International Workshop on Data-Driven Decisions, Experimentation and Evolution (RCoSE/DDrEE)*, pages 16–19, 2019.

[20] N. Munaiah and A. Meneely. Data-driven insights from vulnerability discovery metrics. In *2019 IEEE/ACM Joint 4th International Workshop on Rapid Continuous Software Engineering and 1st International Workshop on Data-Driven Decisions, Experimentation and Evolution (RCoSE/DDrEE)*, pages 1–7, 2019.

[21] Statista The Statistics Portal. Combined desktop and mobile visits to amazon.com from may 2019 to september 2020, 2021.

[22] Statista The Statistics Portal. Number of daily active facebook users worldwide as of 4th quarter 2020, 2021.

[23] Statista The Statistics Portal. Number of netflix paid subscribers worldwide from 1st quarter 2013 to 1st quarter 2021, 2021.

[24] Statista The Statistics Portal. The world's 50 largest companies based on number of employees in 2019, 2021.

[25] Olli Rissanen and Jürgen Münch. Continuous experimentation in the b2b domain: A case study. 03 2015.

[26] Rasmus Ros and Per Runeson. Continuous experimentation and a/b testing: A mapping study. In *RCoSE'18 Proceedings of the 4th International Workshop on Rapid Continuous Software Engineering*, pages 35–41, United States, 2018. Association for Computing Machinery (ACM). RCoSE 2018 : 4th International Workshop on Rapid Continuous Software Engineering ; Conference date: 28-05-2018 Through 28-05-2018.

[27] Graeme Ruxton. The unequal variance t-test is an underused alternative to student's t-test and the mann-whitney u test. *Behavioral Ecology*, 17, 04 2006.

[28] Martin Saveski, Jean Pouget-Abadie, Guillaume Saint-Jacques, Weitao Duan, Souvik Ghosh, Ya Xu, and Edoardo M. Airoldi. Detecting network effects: Randomizing over randomized experiments. KDD '17, page 1027–1035, New York, NY, USA, 2017. Association for Computing Machinery.

[29] Margaret-Anne Storey, Emelie Engstrom, Martin Höst, Per Runeson, and Elizabeth Bjarnason. Using a visual abstract as a lens for communicating and promoting design science research in software engineering. In *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 181–186, 2017.

Appendices

Appendix A
About This Document

The following environments and tools were used to create this report:

- Operating system: Mac OS Big Sur 11.1
- Visual Studio Code
- Google Analytics
- Google Tag Manager

Appendix B
Interview Questions

Questions for Interview with Product owner, Manager and Developers

1. Context
   The purpose of this interview is to bring a qualitative depth into the thesis I am conducting on introducing A/B testing in CA, your internal tool for IKEA co-workers to interact with customer data.
   The first step in an A/B test is to define goals and appropriate metrics to measure the level of reach towards these goals. Therefore I've done some background research on what data is currently available in CA. But in order to fully grasp the actual objectives and find efficient metrics, I am conducting interviews with a variety of stakeholders that are involved with CA. Since you have had a great impact in the design of customer admin, I believe your knowledge could bring valuable insights to my project. Would you be comfortable with my recording this interview?

2. Introducing questions
   - What is your current role at IKEA digital?
   - What are your responsibilities and daily tasks?
   - What is your involvement with CA?

3. Main questions
   a. Refining goals
   - What is the overall goal that Customer Admin serve in your opinion? How is that valuable to IKEA from a business perspective?
   - What are some of your most active users doing in CA? What satisfies those users? How can you encourage other users to be as active?
   - In your opinion, how is the user ensured the change is successfull, do you think it is clearly indicated that the change was saved successfully?
   - What are the users of CA's biggest pain points? Do pain points vary across markets?
   - Would you say that time spent is a critical success factor for the user experience? How long does it usually take?
   - Are there any complaints and in that case what type of complaints do you hear in your support, or through user research?

   b. Defining goals
   - What do you believe is good for your IKEA coworkers when using internal tools like CA?
   - Where are the biggest opportunities for improvement upon the user experience in CA?
   - Where would you prefer to spend time and efforts making an impact?

   c. Metrics
   - What is the desired outcome of deploying CA?
   - How do you expect to measure the outcome of your work? Would you say that you and your colleagues are all aligned? Is it possible to efficiently measure a change in these metrics?

- How are the desired outcome and its related measure connected to IKEA Digital's goals? Do you believe these metrics will have a meaningful effect on business?
- Can you think of any secondary metrics? That is metrics that does not have a direct impact on the desired outcome but will have side effects on the user experience.

4. Summary

   I believe we have covered all questions that I had in mind and that I think will guide me in defining goals for an A/B test. However, is there anything you would like to add to these questions?

   I will conduct some other interviews as well, compile the answers and incorporate it into the project and my report. This will comprise the basis for how goals and metrics are defined in the A/B test. As follow up on these interviews I will share my work, which will be finalized before the 6th of June.

Questions for Interview with Users

5. Context
   The purpose of this interview is to bring a qualitative depth into the thesis I am conducting on introducing A/B testing, a software testing methodology, in CA, your internal tool for IKEA co-workers to interact with customer data.
   The first step in such a test is to define goals and appropriate metrics to measure the level of reach towards these goals. In order to fully grasp the actual objectives and find efficient metrics, I am conducting interviews with a variety of stakeholders that are involved with CA. Since you have a user perspective of customer admin, I believe your knowledge could bring valuable insights to my project.
   Would you be comfortable with my recording this interview?

6. Introducing questions
   - What is your current role at IKEA digital?
   - What are your responsibilities and daily tasks?
   - What is your involvement with CA?

7. Main questions
   a. Refining goals
      - What is most usually your agenda when opening CA? How do you go about to do this? Why do you go this way?
      - How does CA help you accomplish your agenda?
      - How do you ensure yourself that you have completed the task successfully?
      - In case you don't succeed, what is usually the struggle and how do you discover things didn't go as planned?
      - Would you say that time is a critical success factor for your user experience?
      - What are your biggest pain points when using CA?
      - Are there any complaints you hear among your colleagues?

   b. Defining goals
      - What do you believe is good for you and your colleagues when using CA?
      - Where are the biggest opportunities for improvement in CA?

8. Summary
   I believe we have covered all questions that I had in mind and that I think will guide me in defining goals and metrics for an A/B test. However, is there anything you would like to add to these questions?
   I will conduct some other interviews as well, compile the answers and incorporate it into the project and my report. This will comprise the basis for how goals and metrics are defined in the A/B test. As follow up on these interviews I will share my work, which will be finalized before the 6th of June.

Appendix C
Questionnaire

Questionnaire – A/B test in Customer Admin

The Customer Engagement team has conducted an A/B testing study in the Customer Admin tool, where you are one of the participants!
The test launches two different versions of Customer Admin (the original version and a new version) and based on behavioral data we can evaluate what version performs best. However, data doesn't say it all, so therefore we are interested in your personal opinions and feedback. This questionnaire is anonymous and will help us understand our users and how to develop Customer Admin.

1. Rate the user friendliness of the original CA version, i.e. the old menu design?

Mark only one oval.

	1	2	3	4	5	
Not good	◯	◯	◯	◯	◯	Very good

2. What makes the user friendliness good/bad in the original?

3. Rate the aesthetics of the original CA version, i.e. the old menu design?

Mark only one oval.

	1	2	3	4	5	
Not good	◯	◯	◯	◯	◯	Very good

4. What makes the aesthetics good/bad in the original?

5. Rate the user friendliness of the new CA version, i.e. the new menu design?

Mark only one oval.

	1	2	3	4	5	
Not good	◯	◯	◯	◯	◯	Very good

6. What makes the user friendliness good/bad in the new version?

7. Rate the aesthetics of the new CA version, i.e. the new menu design?

Mark only one oval.

	1	2	3	4	5	
Not good	◯	◯	◯	◯	◯	Very good

8. What makes the aesthetics good/bad in the new version?

9. What version do you prefer?

Mark only one oval.

- ◯ Original version with the old menu design
- ◯ New version with the new menu design

10. Has your participation in the A/B test affected your work and ability to manage customer errands? If yes, why?

11. Has your participation in the A/B test affected you in any other way? If yes, how?

Can A/B testing enhance the development of internal tools?

Today companies all over the world use various internal tools in order to manage and streamline their operations. These tools are often used continuously, some on a daily basis, by co-workers. Thus there is a need to consecutively develop these tools in order to keep up with evolving organizations. A/B testing is recognized for providing valuable feedback to software developers of commercial tools, guiding them on how to make the software better for the user. This begs the question – could A/B testing be found as valuable for the development of internal tools?

A/B testing is recognized as a test method that allow software developers to quickly evaluate new features and design ideas through real world user feedback. Essentially a selected group of users are randomly assigned to one of two versions: the control (A) version which is usually the original version, and the treatment (B) version which contains the new feature or design element of interest. Metrics are thereafter collected, embodying the performance of each version in the light of a predetermined desired impact on user behavior. Finally statistics are derived, determining which level of significance the versions differ.

A/B tests that are conducted today are in the external domain and major internet companies such as Google, Microsoft and Amazon are leading the development running tens of thousands concurrent tests. However, alongside the digitalization, companies have not only transformed the customer interaction but moreover digitized and digitalized information throughout value chains, through tools such as inventory management systems or other administration systems. Therefore A/B testing internal tools is likewise relevant and should be empirically evaluated as a solution instance on the development of internal tools.

To do this, I have implemented an A/B test as a proof of concept on Customer Admin, IKEA's internal tool that allow co-workers to interact with customer data. Firstly interviews were held with various stakeholders which laid the groundwork for forming a hypothesis with a key metric that addressed the actual goal of a new design element. Thereafter an A/B test was launched and data was collected for 33 days. As a complement to the statistics, user questionnaires were sent out in order to involve user preferences in the solution validation.

Due to low amounts of data, the hypothesis could not be accepted on a sufficient level of significance. Other secondary metrics, however, suggested one version's superiorness and gave insights into the users behavior.

The scarce data was found to be a consequence of traffic being relatively low which in many companies can be generalized on internal tools. In conclusion A/B tests on internal tools are unlikely to reach the same level of significance as A/B tests of external websites and services. Measures on how to minimize this defect was recommended, e.g. specify frequently triggered metrics and testing larger user groups.

Amalia Paulsson
Faculty of Engineering (LTH), Lund University